

硕士学位论文

基于开源有限元的水凝胶力学行为研究

**STUDY ON MECHANICAL BEHAVIOR OF
HYDROGELS BASED ON OPEN SOURCE FINITE
ELEMENT CODE**

章肖阳

哈尔滨工业大学
2020 年 6 月

国内图书分类号：O343.5

学校代码：10213

国际图书分类号：531

密级：公开

工学硕士学位论文

基于开源有限元的水凝胶力学行为研究

硕 士 研 究 生：章肖阳

导 师：洪伟教授

联 合 导 师：肖思博士

申 请 学 位：工学硕士

学 科：航天工程

所 在 单 位：南方科技大学

答 辩 日 期：2020 年 6 月

授予学位单位：哈尔滨工业大学

Classified Index: O343.5
U.D.C: 531

Dissertation for the Master's Degree of Engineering

**STUDY ON MECHANICAL BEHAVIOR OF
HYDROGELS BASED ON OPEN SOURCE FINITE
ELEMENT CODE**

Candidate:	Zhang Xiaoyang
Supervisor:	Prof. Hong Wei
Associate Supervisor:	Dr. Xiao Si
Academic Degree Applied for:	Master of Engineering
Speciality:	Aerospace engineering
Affiliation:	Southern University of Science and Technology
Date of Defence:	June, 2020
Degree-Conferring Institution:	Harbin Institute of Technology

摘 要

本课题的内容主要围绕聚合物凝胶溶胀的非平衡态热力学理论模型展开，并针对凝胶在溶胀和退溶胀过程中模量随高分子体积分数变化的标度律加以修正，利用开源有限元平台结合上述理论研究了一些有趣的凝胶溶胀现象。

对于化学凝胶溶胀的非平衡态热力学理论，回顾介绍了经典的化学凝胶溶胀理论，在此基础上推导出有限元建模的控制方程，并在开源平台 FEniCS 上实现了凝胶溶胀仿真程序。其次，本课题还将最近发表的水凝胶溶胀与退溶胀过程中的标度律扩展到描述复杂构型凝胶结构在力化学耦合作用下行为的连续介质场论，并进行了有限元实现。

将开源有限元平台 FEniCS 应用于研究上述化学凝胶溶胀的非平衡态热力学理论，利用上述推导出有限元建模的控制方程建立相关的 FEniCS 化学凝胶溶胀模型，并在上述程序的基础上进行扩展应用：将标度律修正的水凝胶场论应用到化学凝胶溶胀 FEniCS 二维模型中，与经典 Flory-Huggins 理论以及 COMSOL 模型的结果进行比较；结合 FEniCS 建模研究水凝胶薄膜溶胀过程中的起皱现象；结合 FEniCS 建模研究不对称截面化学凝胶长条溶胀动力学行为。

FEniCS 非常适合处理多物理场耦合的有限元问题，相比商用软件，其主要优势是开放性、支持高性能并行运算、并内置多种求解器，开源自定义的形式可以让研究者对所研究问题的理论有更深刻的认识，非常适合用于教学与科研。

在过去的十年中，大量的研究人员致力于提高水凝胶性能，使其在生物医学、柔性电子、传感器和致动器等众多领域的应用机会大大增加。研究水凝胶的溶胀的理论模型以及基于理论的数值模型与仿真工具对扩展水凝胶的应用而言非常重要。此研究将不同领域的先进技术结合起来，致力于提高水凝胶的力学性能、设计制造更实用的结构，引进一些新的功能，并结合其工程实际应用，推动水凝胶研究的进一步创新发展。

关键词：水凝胶；有限元；FEniCS；标度律；褶皱

Abstract

This study mainly focuses on the field theory of polymeric gels based on non-equilibrium thermodynamics, and a modification by considering a recently proposed scaling law between the modulus and polymer volume fraction during swelling and deswelling. A series of interesting phenomena of hydrogels and gel structures have been studied by implementing the field theory on an open source finite element platform, FEniCS.

This thesis first reviews the non-equilibrium thermodynamics of gel swelling and the derivation of the governing equation for the continuum field theory, which was used to implement a finite element code on FEniCS. Secondly, a recently proposed scaling law between the elastic modulus and the polymer volume fraction of swelling and deswelling hydrogels is extended into a free energy function, and then implemented into the finite element code.

The open source finite element platform, FEniCS, was applied to study the swelling and deformation of polymeric gels. Several examples are used to verify the numerical code and to demonstrate the applications of the model: (1) swelling and deswelling process by applying a model consistent with the scaling law; (2) the wrinkling instability in a double-layer gel structure; (3) the kinetics-induced peculiar bending phenomenon of a 3D-printed hydrogel strip with asymmetric section.

FEniCS is suitable for multi-physics multi-field coupled problems. Major advantages of FEniCS include the support of high-performance parallel computation and a variety of built-in solvers. The open-source platform and user-friendly language allow researchers to focus more on the physical problems, and are very suitable for research as well as teaching.

Over the past decade, a large number of researchers have focused on improving the mechanical performance of hydrogels, and developing applications in biomedical, flexible electronics, sensors and actuators. The theoretical studies on the behaviors of hydrogels and physics-based models and numerical tools are essential to the development of these applications. Scientists are committed to improving the mechanical properties of hydrogels, designing and manufacturing more practical structures, introducing some new functions, and to promoting engineering applications of hydrogels by combining the advanced technologies from different fields.

Keywords: hydrogel, finite element, FEniCS, scaling law, wrinkle

目 录

摘 要.....	I
ABSTRACT.....	II
第 1 章 绪 论.....	1
1.1 课题的来源及研究目的和意义.....	1
1.2 开源有限元平台.....	1
1.3 国内外研究现状分析.....	3
1.3.1 国外开源有限元平台研究现状.....	3
1.3.2 国内研究现状.....	4
1.4 课题主要研究工作.....	5
第 2 章 超弹性与水凝胶.....	6
2.1 超弹性体变形理论简介.....	6
2.2 水凝胶简介.....	7
2.3 水凝胶分类.....	9
2.3.1 根据交联方式分类.....	9
2.3.2 水凝胶的其他分类.....	11
2.4 水凝胶的合成与特性.....	13
2.5 水凝胶的应用.....	14
2.6 水凝胶在力学方面当前研究的主要方向.....	15
2.6.1 提高水凝胶的机械性能.....	15
2.6.2 水凝胶的断裂和自愈.....	16
2.6.3 水凝胶致动器.....	17
2.6.4 搭建复杂的结构.....	18
2.7 本章小结.....	19
第 3 章 水凝胶理论模型.....	20
3.1 化学凝胶溶胀理论.....	20
3.1.1 水凝胶的溶胀与大变形理论.....	20
3.1.2 水凝胶的溶胀控制方程推导.....	24
3.2 水凝胶含水量对弹性模量的影响.....	28
3.3 与 COMSOL 仿真以及实验结果对比.....	32
3.3.1 标度律修正的溶胀模型仿真结果.....	33
3.3.2 标度律修正的退溶胀仿真结果.....	34
3.4 化学凝胶溶胀试验.....	35

3.4.1 凝胶拉伸试验.....	35
3.4.2 理论推导.....	36
3.5 本章小结.....	37
第 4 章 基于 FEniCS 的水凝胶力学行为研究.....	39
4.1 FEniCS 简介及简单示例.....	39
4.1.1 FEniCS 开源有限元平台简单介绍.....	39
4.1.2 FEniCS 程序示例.....	40
4.2 水凝胶的溶胀 FEniCS 程序.....	44
4.2.1 搭建 FEniCS 水凝胶溶胀模型.....	44
4.2.2 网格尺寸的收敛性分析:	46
4.2.3 时间步长的收敛性分析.....	47
4.2.4 FEniCS 水凝胶溶胀混合单元模型.....	49
4.3 FEniCS 化学凝胶溶胀力学行为研究.....	51
4.3.1 基于标度率修正的化学凝胶溶胀模型.....	51
4.3.2 水凝胶薄膜的起皱现象.....	52
4.3.3 三维打印凝胶长条溶胀行为研究.....	55
4.4 本章小结.....	58
结 论.....	59
参考文献.....	60
附录 A 文中 FEniCS 计算涉及到的 Python 源代码.....	69
A.1 FEniCS 程序中变量与正文中公式对应变量对照表.....	69
A.2 FEniCS 常用 Python 函数对象.....	72
A.2 FEniCS 常用求解器以及预处理.....	77
附录 B 文中 FEniCS 计算涉及到的 Python 源代码.....	80
B.1 FEniCS 二维化学凝胶溶胀模型.....	80
B.2 FEniCS 二维化学凝胶溶胀混合单元模型.....	83
B.3 针对标度律修正的化学凝胶溶胀模型.....	87
B.4 针对标度律修正的化学凝胶退溶胀模型.....	91
B.5 水凝胶薄膜的起皱模型.....	95
B.6 化学凝胶长条溶胀模型.....	101
攻读硕士学位期间发表的论文及其它成果.....	109
哈尔滨工业大学与南方科技大学联合培养研究生学位论文原创性声明和使用权限	110
致 谢.....	111

第 1 章 绪 论

1.1 课题的来源及研究目的和意义

软物质物理是当前材料领域的研究热点之一，特别是聚合物凝胶、超强超韧凝胶和柔性复合材料的理论模型构筑。水凝胶是一种特殊的软湿性材料，兼有固体和液体性质。它是由聚合物网络和大量的溶剂小分子等组成的三维网络材料^[29]。这些三维网络结构通常通过化学、物理交联形成。水凝胶已经广泛应用于生物医学、柔性电子、传感器和仿生驱动器各个领域。

传统水凝胶机械强度通常较低，而且容易发生永久性的破坏，这大大限制了水凝胶在工程上的应用。促进水凝胶在工程领域的应用和发展需要改进水凝胶的物理化学性质，引入动态研究调节机制以及设计更复杂的水凝胶结构^[30]。

研究水凝胶的溶胀的理论模型以及影响其力学性能的因素对于水凝胶的应用非常重要。今后的水凝胶研究的发展趋势是将不同领域的先进技术结合起来，致力于提高水凝胶的力学性能、设计制造更实用的结构，引进一些新的功能，还应该与工程实际应用相结合。

1.2 开源有限元平台

有限元方法对现代计算力学的发展极为重要。ANSYS、ADINA、ABAQUS、MSC、COMSOL 是目前主流的几款商业有限元软件，其中 ADINA、ABAQUS 在非线性分析方面有较强的能力，目前是业内较为认可的两款有限元分析软件，ANSYS、MSC 进入中国比较早，所以在国内知名度高、应用广泛，COMSOL 中集成了强大的多物理场模块以及用户自定义功能，近年来受到众多研究人员的青睐。

开源有限元平台与商业软件相比有一些自己的优势。常见的开源有限元平台有 MOOSE, Deal.II, FEniCS, OOFEM, GetFEM, DUNE, GRINS, FreeFem++, OpenFOAM, DOLFIN 等。中国科学院数学与系统科学研究所梁国平研究员团队也建立了开放式的有限元程序自动生成系统 FEPG^[1]。虽然 ABAQUS 等商业有限元软件能够解决复杂和深入的工程问题，并且具有强大的非线性分析功能，但是一些开源有限元平台在多物理场耦合计算、高性能并行计算方面具有一定的优势。并且开源有限元平台的高度用户自定义特性使其适用的领域更加广泛，在解决有限元问题的过程中能够获得问题的全部源码，有利于深入学习以及用于教学，借助现有框架能够更加便捷的进行开发。这些开源有限元平台比较适合开发者深入了解所研究的问题以及有限元方法。笔者根据自己的经验总结了其中三种开源有

有限元平台 MOOSE, Deal.II, 和 FEniCS 功能上的差异, 如表 1-1 所示, 并将三个平台各自的优缺点分述如下。

MOOSE (Multiphysics Object-Oriented Simulation Environment 面向对象的多物理场仿真环境) 通过利用最先进的全耦合、全隐式多物理求解器, 同时提供自动并行化、网格自适应和不断完善物理模块, 帮助开发应用程序^[2, 3]。它的优点有: (1)MOOSE 具有庞大且不断完善的物理模块集, 包括固体力学、相场、流体力学和导热等; (2)MOOSE 支持多尺度模型, 支持基于 MOOSE 的应用程序的链接, 支持跨应用程序、时间尺度和空间域的协作; (3)MOOSE 能够进行高性能并行计算; (4)结合 peacock, 使其拥有强大的后处理能力和优美的图形界面。缺点: 封装度较高, 二次开发难度较大。

表 1-1 三种不同有限元平台功能比较

名称	MOOSE	Deal.II	FEniCS
模块完整度	★★★★★	★★	★★★
并行运算能力	✓	✓	✓
多尺度建模	✓	✓	✓
网格划分能力	★★	★★★★★	★★★
外部应用扩展	★★★★★	★★	★★★★★
后处理能力	★★★★★	★★★★	★★★★★
图形界面	✓	✓	✓
使用难度	★★★	★★★★★	★
二次开发难度	★★★★★	★★★★	★★★
指导文件完整度	★★★	★★★★★	★★★★★

Deal.II 是一个 C++程序库, 针对偏微分方程使用自适应有限元计算解的方法, 提供了所需的复杂数据结构和算法的现代接口^[4, 5]。Deal.II 的主要目的是使现代有限元代码的快速发展成为可能, 在其他方面使用自适应网格和广泛的工具类在有限元程序中经常使用。优点: (1)优秀的网格处理和细化处理; (2)优秀的自由度处理、网格输入和图形格式输出能力; (3)能够进行高性能并行计算; (4)结合 Paraview, 使其拥有强大的后处理能力和优美的图形界面; (5)互联网主页提供每一个例子的详细教程, 包括各种线性求解器的编写, 无旋流动, 边界元方法, 超弹性问题等复杂问题, 非常适合有理论背景的开发者的学习。缺点: (1)模块较少, 功能不完善; (2)每个例子都需要编写完整程序, 学习和使用难度较大; (3)可移植性和适应性差。

FEniCS 是另一个用于求解多物理场耦合问题的开源有限元计算平台，能够使用户能够快速地将科学模型转换为高效的有限元代码。FEniCS 的特点是在 Python/C++ 语法基础上，进一步针对有限元的数学语言进行了语法抽象，从而可以用一种非常贴近数学语言的方式来编写有限元程序^[6, 7]。FEniCS 拥有一套自己的有限元语言的编译器，从而可以自动生成底层代码。由于用户在非常高的抽象层次上进行编程，所以不需要考虑很多底层的处理。例如矩阵组装，数值积分，线性代数处理等等。FEniCS 运行在多种平台上，具有以下优点：(1) 入门比较容易；(2) 功能和模块比较完善；(3) 能够进行高性能并行计算；(4) 结合 Paraview，使其拥有强大的后处理能力和优美的图形界面；(5) 官方提供了相当丰富的指导书籍及其他说明文档。FEniCS 的主要缺点是二次开发相对较困难。

1.3 国内外研究现状分析

1.3.1 国外开源有限元平台研究现状

MOOSE: 瑞典皇家理工学院的 Brandt 和 Djurfeldt 在 2015 年利用 MOOSE 和 NeuroRD 模拟大规模神经科学中的多尺度多物理问题，在 MOOSE 中模拟了体细胞，树突和脊柱的简单隔室模型的电气特性， Ca^{2+} 在脊柱中的依赖性反应和扩散是在 NeuroRD 中模拟的^[10]；美国爱达荷国家实验室的 Peterson 和 Kong 在 2018 年利用 MOOSE 的 Navier-Stokes 模块解决了不可压缩的 Navier-Stokes(INS)方程，并做了图形化的软件和回归、验证测试并整理了相关的综述^[11]。

耦合非线性偏微分方程系统经常出现在核工程的模拟中，MOOSE 可以为此类问题提供解决方案。美国爱达荷国家实验室多物理方法组的 Gaston 和法国格勒诺布尔技术研究所的 Lebrun-Grandié 在 2009 年对利用 MOOSE 解决上述问题并做了相关综述，并举了相关例子，例如：简单反应堆的中子通量模拟；全隐式全耦合反应器燃料性能仿真(BISON)；鹅卵石床反应堆模拟(Pronghorn)^[12]；加拿大皇家军事学院的 Gamble 和 Wowk 2015 年首次利用 MOOSE 框架模拟 CANDU 反应堆（加拿大原子能公司、安大略水电公司和加拿大奇异公司联合研发的压水式重水反应堆）的燃料元件三维变形，发现粘性，无摩擦接触算法的运动学和 penalty 函数模型计算在二维模型和三维模型计算中都表现正确；他们还利用 MOOSE 框架研究多物理场复杂条件下反应堆燃料组件热弯曲行为，计算了各个燃料元件组件的抗弯刚度和热弯曲度，结果与解析解契合^[13]。

Deal.II: 自 Deal.II 项目启动以来，已经成功地应用于解决学术研究、应用科学和工业项目领域中的各种各样的问题，如有限元离散化的数学误差分析；不可压缩和可压缩流体流动；流体结构相互作用；生物医学逆成像问题等^[14]。

在可压缩流体方面：海德堡大学应用数学研究所的 Hartmann 和 Houston 在

2017 年采用 Deal.II 基于可压缩 Navier-Stokes 方程的不连续伽辽金离散化来研究空气动力学^[15]；然后 Hartmann 又利用 Deal.II 给出了超音速可压缩流动和与压力诱导阻力系数相关的伴随解。

在生物医学成像方面：德州 A&M 大学光子迁移实验室的 Joshi 和德克萨斯大学 ICES 地下建模中心的 Bangerth 在 2010 年利用 Deal.II 编写了能够实现完全自适应的三维生物医学成像方法和其它成像方法的程序，能够将实验获得的光学断层扫描数据转换成肿瘤位置、形状和大小的三维图像^[16]。

FEniCS：在固体力学方面，2015 年伦敦帝国学院航空工程系的 Phunpeng 和 Baiz 使用 FEniCS 环境求解应变梯度弹性问题，将得出的数值结果与著名的基准算例进行了比较，验证了 FEniCS 在这类应用中的适用性^[17]；2018 年，印度理工学院金奈校区集成建模与仿真实验室的 Natarajan 和 Annabattula 使用 FEniCS 来研究准静态脆性断裂问题，提出了二维和三维脆性断裂问题交错相场模型，并对结果进行了验证。结果表明，长度尺度和加载速率的选择对获得准确的计算结果具有重要意义^[18]；汉堡大学数学系的 Abert 和奥地利应用科学大学的 Exl 在 2013 年开发了一个基于 FEniCS 的有限元微磁模拟程序，应用变换方法来解决退磁场问题^[19]。流体力学方面，2012 年多伦多大学机械与工业工程系生物医学模拟实验室的 Valen-Sendstad 和 Steinman 利用 FEniCS 研究血液在动脉瘤中的流动^[20]；2013 年柏林科技大学力学研究所的 Lofink 和 Muller 利用 FEniCS 研究血液在大动脉中的流动，为了克服血流与弹性容器壁相互作用变形建模困难的问题，他们提出一个血液流经动脉血管的二维模型，通过质量守恒和动量守恒作为控制方程、并定义与时间有关的边界条件，利用 FEniCS 有限元平台求解，得出了合理的结果^[21]。

1.3.2 国内研究现状

国内一些课题组近年来也开始使用开源有限元平台解决问题。

早在 2003 年，情报指挥控制系统与仿真技术杂志的施桢华就提出了 MOOSE 与互联网结合，采用分布式模型定义的仿真方法^[22]；哈尔滨工程大学的赵博康和田兆斐教授在 2018 也使用 MOOSE 进行了反应堆多物理场、多尺度模拟工作^[23, 24]。西安理工大学的王晶晶在 2017 年结合 FEniCS 计算当材料有纳米椭圆孔缺陷时，表面效应在此模型下对材料力学性能的影响^[25]。

由上述情况可以看出国内外学者利用开源有限元平台结合各自的研究方向做了十分丰富的工作，有限元方法作为力学分析常用方法正在越来越多的领域发挥自己的价值。

开源有限元平台不仅可以应用于研究脆性断裂问题、梯度应变问题等固体力

学问题；还可以应用于研究流体力学，甚至可以将其应用于医学成像和医疗诊断；除此之外，开源有限元方法还被应用于有限元离散化的数学误差分析；流体结构相互作用；磁流体动力学；生物医学逆成像问题；燃料电池建模；晶体生长甚至反应堆多物理场耦合等复杂问题。

开源有限元的强大用户自定义能力，使其能够为相当广泛领域的问题提供解决方案，包括应用于学术研究、应用科学和工业项目的各种各样的问题中。固体力学，流体力学，热力学，化学，空气动力学以及更复杂的跨尺度、多物理场问题，生物、医学等方面的问题都可以结合开源有限元平台，通过这样的方法不仅可以获得更加合适的解决方案，更能深入了解该领域的机理，推进我国基础科学，应用科学的发展。

1.4 课题主要研究工作

本课题的内容主要围绕化学凝胶溶胀的非平衡态热力学理论模型、水凝胶在溶胀退溶胀过程中模量随高分子体积分数变化的标度律展开；利用开源有限元平台 FEniCS 结合上述理论研究了很多凝胶溶胀力学现象。课题中完成的工作如下：

(1) 回顾介绍了经典的力化学耦合凝胶溶胀模型和场论^[8]，并根据上述理论推导出有限元建模的控制方程，用于实现聚合物凝胶溶胀的有限元代码。

(2) 用最新提出的标度律^[9]对凝胶的自由能函数进行修正，扩展耦合溶胀变形模型，并实现对应的有限元代码。利用上述修正模型推导解释了使用不同浓度的有机溶液配制组分相同的水凝胶，使不同凝胶样本吸水溶胀到高分子体积分数相同时，力学性能不同的现象。

(3) 学习使用开源有限元平台 FEniCS，根据化学凝胶溶胀的非平衡态热力学理论，编写了化学凝胶溶胀二维模型的 FEniCS 程序。选择合适的网格尺度和时间步长的模型，将运算结果与商用软件 COMSOL 的结果比较验证。在上述基础上进一步编写了使用混合单元研究化学凝胶溶胀二维模型的 FEniCS 程序。

(4) 利用上述凝胶溶胀模型的 FEniCS 程序研究以下应用：基于标度律的水凝胶溶胀退溶胀现象；水凝胶薄膜溶胀过程中的起皱现象；三维打印凝胶长条的异常溶胀动力学行为；

(5) 在论文的附录部分将会提供作者在学习 FEniCS 开源有限元平台过程中的一些总结和心得，包括：如何将 Abaqus 中绘制好的网格模型转换成 FEniCS 能够识别的.xml 文件；FEniCS 常用 Python 函数对象简介及其使用方法；FEniCS 常用求解器以及预处理的简介及使用方法，以及相应的源代码。

第 2 章 超弹性与水凝胶

2.1 超弹性体变形理论简介

水凝胶是由聚合物网络和大量的溶剂小分子等组成的三维网络材料^[29]。这些三维网络结构通常通过化学、物理交联形成。在等温过程中物体发生变形时，物体的热力学状态仅依赖于其应变 \mathbf{E} ，这样的物体称为超弹性体，或者称为 Green 弹性体。水凝胶材料本构通常被认为是超弹性。

超弹性体的本构关系可以用自由能函数来表示，由 Gibbs 方程^[26]：

$$\rho_0 \theta d\eta = \rho_0 d\varepsilon - \mathbf{T} : d\mathbf{E} \quad (2-1)$$

式中 ρ_0 —— 材料密度 (kg/m^3)；
 θ —— 绝对温度；
 η —— 单位质量上的熵密度；
 ε —— 内能；
 \mathbf{E} —— 应变；
 \mathbf{T} —— 与 \mathbf{E} 功共轭的应力。

当 \mathbf{E} 为 Green 应变时， \mathbf{T} 就是第二类 Piola-Kirchhoff 应力^[26]。

在等温过程中物体发生变形时，物体的热力学状态仅依赖于其应变 \mathbf{E} ，此时：

$$\mathbf{T} : d\mathbf{E} = \rho_0 (d\varepsilon - \theta d\eta) = \rho_0 d(\varepsilon - \theta\eta) = \rho_0 d\psi = \rho_0 \frac{\partial \psi}{\partial \mathbf{E}} : d\mathbf{E} \quad (2-2)$$

ψ 为 Helmholtz 自由能，其可作为势函数，满足：

$$\mathbf{T} = \rho_0 \frac{\partial \psi}{\partial \mathbf{E}} \quad (2-3)$$

势函数可以写成伸长比 λ 的函数 $W(\lambda)$ ，也可以写成应变主不变量 $I_1 I_2 I_3$ 的函数 $\psi(I)$ 。常用的超弹性体模型如下^[27]：

新胡克模型：

$$\psi = \frac{\mu_1}{2} (I_1 - 3) \quad (2-4)$$

式中 μ_1 —— 剪切模量（材料常数）；
 I_1 —— 右柯西格林张量的第一主不变量^[26]。

Mooney-Rivlin 模型：

$$\psi = \frac{\mu_1}{2} (\bar{I}_1 - 3) + \frac{\mu_2}{2} (\bar{I}_2 - 3) \quad (2-5)$$

式中 μ_1, μ_2 —— 剪切模量（材料常数）；

I_1, I_2 —— 与右柯西-格林张量的主不变量 I_1, I_2, I_3 之间有如下关系^[26]:

$$\begin{aligned}\bar{I}_1 &= J^{-2/3} I_1, I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \\ \bar{I}_2 &= J^{-4/3} I_2, I_2 = \lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_3^2 \lambda_1^2\end{aligned}$$

Gent 模型:

$$\psi = -\frac{\mu J_m}{2} \ln \left(1 - \frac{I_1 - 3}{J_m} \right), J_m = I_m - 3 \quad (2-6)$$

式中 μ —— 剪切模量 (材料常数);

I_m —— 左柯西-格林变形张量的第一主不变量达到一个极限值, 即当 $I_m \rightarrow \infty$, Gent 模型退化为新胡克模型^[26]。

Yeoh 模型:

$$\psi = \sum_{i=1}^3 C_i (I_1 - 3)^i \quad (2-7)$$

式中 C_i —— 材料常数, $2C_1$ 可以解释为初始剪切模量^[26]。

Arruda-Boyce 模型:

$$\psi = C_1 \left[\begin{aligned} &\frac{1}{2}(I_1 - 3) + \frac{1}{20\lambda_m^2}(I_1^2 - 9) \\ &+ \frac{11}{1050\lambda_m^4}(I_1^3 - 27) + \frac{19}{7000\lambda_m^6}(I_1^4 - 81) + \dots \end{aligned} \right] \quad (2-8)$$

式中 C_1 —— 剪切模量 (材料常数);

λ_m —— 高分子网络被锁住不能拉伸时的拉伸比^[26]。

当超弹性体不可压时, 有 $\det \mathbf{F} = 1$, 该约束条件可以通过拉氏乘子法添加到超弹性体的势函数中^[27]:

$$\psi(\lambda) = \psi(\lambda_1, \lambda_2, \lambda_3) - \Pi(\lambda_1 \lambda_2 \lambda_3 - 1) \quad (2-9)$$

式中 Π 是拉氏乘子, 此时对应的应力应变关系如下:

$$\sigma_i = \lambda_i \frac{\partial \psi(\lambda)}{\partial \lambda_i} - \Pi \quad (2-10)$$

拉氏乘子并不是材料参数, 它在公式 (2-10) 中的物理意义类似于静水压。

2.2 水凝胶简介

水凝胶是一种兼有固体和液体性质的独特凝聚态材料, 它是由聚合物网络、小分子和大量的溶剂等组成的三维网络材料^[29]。这些三维网络结构通常通过化学、物理交联形成。亲水基团在水凝胶交联中起着重要的作用。亲水基团如羟基 (-OH)、羧基 (-COOH) 和磺基 (-SO₃H) 等官能团可以通过物理或化学键将聚合物链交联^[29]。水凝胶在密度上更接近于液体, 并且它的两个重要特性就是相态和流变

特性，但是水凝胶通常被当作固体来处理。对于物理凝胶，聚合物网络之间的交联是通过各种各样较弱的相互作用，如聚合物链缠绕、氢键或范德华力等。这样的交联形成的结构通常稳定性较差，当它们浸泡在溶剂中时，凝胶会随着时间的推移而溶解。聚合物链也可以通过化学反应交联，形成更强的共价键。通过化学方式交联的网络更加稳定。因此，在大多数的应用领域中，化学凝胶通常是较好的选择。很多聚合物网络都能在溶剂中溶胀形成凝胶，不同的聚合物凝胶有不同的性质和用途，它们在工业上有大量的应用。

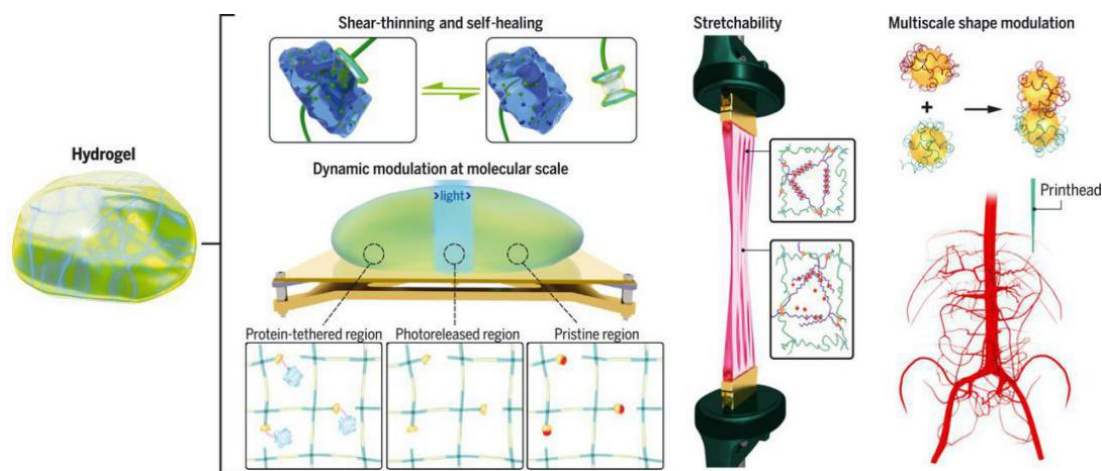


图 2-1 力学性能增强的工程水凝胶：人们在改进水凝胶的机械性能以及使其剪切变薄、自愈和响应性方面取得了进展，还开发了技术来控制水凝胶的形状和结构^[30]

水凝胶的聚合物网络有的是自然界中存在的天然聚合物，其中最具代表性的是多糖。还有一些聚合物网络则是人工合成的。天然聚合物的工业需求量很大，而且以每年 7.1% 的趋势增长^[29]。此外，它们的低毒和良好的生物降解性也使得研究者对天然高分子材料的应用充满了兴趣。如以壳聚糖、海藻酸盐、葡聚糖、淀粉、果胶、纤维素、木质素等的天然高分子聚合物为原料的水凝胶和微纳颗粒等，在生物医学和其他方面都有很大的应用潜力。另一方面，各种能够形成凝胶的合成聚合物也具有不同的工业应用，如聚丙烯酰胺和聚乙烯醇基凝胶。天然聚合物和合成聚合物凝胶均可应用于健康科学领域，如药物控释、持续释药、靶向释药，以及各种新型药物释放系统在水净化领域的应用。聚合物凝胶得益于其独特的特性，已经成为 21 世纪新型先进智能材料不可缺少的一部分，在生物医学、电子、环境、分离技术、食品包装、农业、药学等领域都展现出广阔的应用前景。

在 19 世纪，第一个关于水凝胶的记录是 Graham 在实验中将溶胶凝胶化来制作硅胶^[31]。20 世纪，大量直接或间接的水凝胶研究促进了水凝胶的理论和应用的发展。1936 年，杜邦公司^[32]合成了第一个合成聚合物（聚甲基丙烯酸-2-羟乙酯）

(pHEMA)。Flory, Huggins 以及 Treloar 在 1940 年至 1950 年之间对高分子进行了大量的研究^[33, 37], 为水凝胶的发展奠定了理论基础。Treloar 以研究聚合物网络的弹性而闻名, 而 Flory 则通过对高分子进行实验和理论研究, 促进了现代聚合物科学的发展, 并因此在 1974 年获得了诺贝尔奖。1960 年, Wichterle 和 Lim 在生物医学领域首次报道了水凝胶的应用, 他们将 pHEMA 水凝胶应用于制作隐形眼镜, 开辟了生物医学水凝胶研究的热潮^[38]。水凝胶在分离技术中也有着广泛的应用, 聚合物网络中大量的多孔空间不仅可以用来吸收水分, 并且能够允许溶质通过基体进行选择性的扩散。在 1980 年, 凝胶在法国和德国被应用在尿布产品中。基于上述特点, 水凝胶还能应用于膜分离过程^[39]。近年来, 合成高分子水凝胶在分离技术中的合成、表征和应用受到了广泛关注。

Brannon-Peppas 等人定义水凝胶是一种高分子网络交联亲水的聚合物结构, 可以吸收和保留大量的水并发生膨胀^[40]。Jones 等人指出水凝胶的性能取决于各种重要的参数, 如聚合物链的亲水性和交联程度^[41]。Ganji 等学者发现水凝胶的保水能力取决于聚合物网络中发现的羟基(-OH)、羧基(-COOH)、亚胺基(-CONH)、酰胺基(-CONH₂)和磺基(-SO₃H)等官能团^[42]。

一些水凝胶会根据外部环境的变化(溶剂成分、温度、pH 值和电场)或在受到刺激时发生形状, 颜色, 体积等变化, 它们也被称为智能水凝胶。过去几年的研究表明, 水凝胶可以用于不同的领域, 如农业、生物材料、医药学和分离技术。最近, 水凝胶在废水处理方面的应用也受到了广泛关注^[43]。

2.3 水凝胶分类

根据水凝胶的聚合物网络的交联方式、来源、高分子的组成、电性、孔径大小, 外观形状以及构型, 水凝胶可以进行如下分类。

2.3.1 根据交联方式分类

根据聚合物网络交联方式的不同, 水凝胶可以分为物理水凝胶和化学水凝胶。物理水凝胶通过高分子链缠绕, 高分子自组装、氢键、范德华力、离子以及静电相互作用等各种物理过程进行交联, 而化学水凝胶则是通过共价键进行交联^[44]。

交联过程就是通过不同的交联方法将聚合物链相互连接, 用以提高水凝胶的机械稳定性。通过改变凝胶中交联的密度, 可以调整多孔结构, 调节溶胀程度, 同时交联的程度和类型也会影响聚合物网络的弹性、粘度和不溶性。交联度较低会产生粘稠可以自由流动的水凝胶, 但同时也会导致水凝胶在溶剂中更容易溶解, 稳定性较低。相反, 过高的交联度会使水凝胶易碎, 但却可以改善其蠕变行为和不溶性^[45]。

物理交联包括高分子链之间的缠绕、氢键、范德华力、离子相互作用以及静

电相互作用,其连接强度比化学交联弱。物理交联凝胶化很容易实现,不需要修改聚合物链,而且很容易实现凝胶态和非凝胶态之间的相互转换。许多天然聚合物,如海藻多糖和蛋白质构成的热驱动水凝胶,在凝胶化过程中,聚合物链的物理缠结随着温度的变化而发生,这种变化通常是由溶解度的改变和刚性的填充聚合物骨架的形成引起的(图 2-2A)^[46, 47]。温度的升高或降低可能导致热凝胶化,其中过渡温度分别定义为较低的临界溶液温度(LCST)和较高的临界溶液温度(UCST),凝胶机制可能会随着特定类型的聚合物而变化。呈 UCST 特性的高分子包括一些天然聚合物,如明胶,以及一些合成聚合物,如聚丙烯酸(PAA),当温度下降到对应的 UCST 以下时,它们就会凝胶化。相反,其他的一些高分子表现出 LCST 的特性,例如合成聚合物(聚异丙基丙烯酰胺)(PNIPAAm),它在温度高于 LCST 时形成凝胶。热响应聚合物的 LCST/UCST 可以通过控制它们的分子量、共聚物的比例或疏水、亲水性段的平衡来调节^[48]。非共价键分子自组装包括氢键、范德华力和疏水相互作用也是物理交联^[49]。高分子在上述交联作用下折叠成具有明确结构和功能的网络。自然界中的例子就有胶原蛋白的分级自组装,胶原蛋白是人体中最丰富的蛋白质(图 2-2 B),其分子中富含脯氨酸或羟脯氨酸,这些分子规则排列形成了胶原蛋白的三螺旋结构,并最终形成胶原水凝胶^[49, 50]。物理交联方式还包括螯合作用或静电相互作用。例如,藻酸盐是一种多糖,它的高分子网络会在某些种类的二价阳离子如 Ca^{2+} 或 Ba^{2+} 作用下以迅速凝胶化,成对的螺旋链会包裹并锁住在其中的离子(图 2-2C)^[51]。天然高分子的复杂结构常常使它们沿长链产生不同程度的静电荷。虽然许多天然聚合物由于羧基(如透明质酸和海藻酸盐)的存在而在中性 pH 值下带负电荷,但当胺基含量较多时(如明胶和壳聚糖)也可能带正电荷。相比之下,合成聚合物能更好地控制静电特性,如聚赖氨酸/聚丙烯酸凝胶 (PLL/PAA)^[52]。物理交联的其中一个例子是阴离子和阳离子之间的离子相互作用,当聚合物上相反的电性将它们结合在一起时,聚阴离子和聚阳离子就能形成共聚物。当含有相反电荷的聚电解质溶液混合时,聚合物链缠结形成复合物,由于电荷相互屏蔽而变得不溶于水(图 2-3D)^[53]。

化学交联是通过共价键连接高分子形成聚合物网络,这种交联方式比物理交联更强且更稳定,在发生溶胀的情况下不会损失三维网络结构。化学交联的方法包括自由基聚合、高能辐射、小分子交联、醛互补和聚合物交联,如缩合反应^[54]。

尽管每一种凝胶化方法都有其对应的优点和局限性,但是使用多成分和多机制已经是改进水凝胶配方的未来发展趋势^[55, 56]该方法合成的水凝胶通常表现出优异的物理化学性质,如显著增强的力学性能、可注射性、自愈性,以及进行动态调节的可能性,下文将进一步讨论。

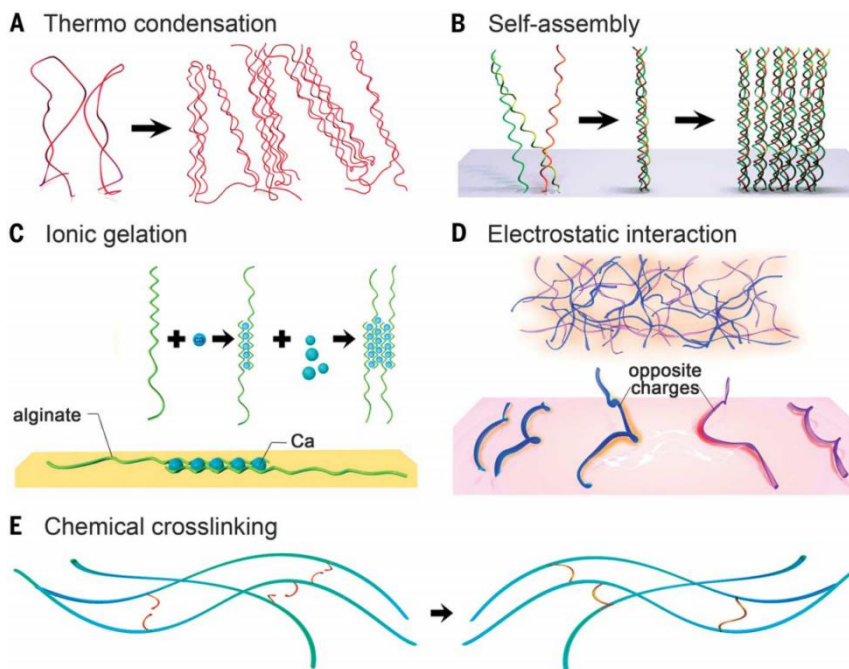


图 2-2 水凝胶的交联方式^[30]

2.3.2 水凝胶的其他分类

(1) 根据高分子网络的来源分类

天然水凝胶：天然聚合物可以自然产生（被称为生物聚合物），最具代表性的是多糖。此外，它们的低毒和良好的生物降解性也引起了研究者对天然高分子材料应用的关注。从天然来源的聚合物如壳聚糖、海藻酸盐、葡萄聚糖、淀粉、果胶、纤维素、木质素等^[57]。

合成水凝胶的三维网络是合成聚合物，由人造单体通过化学聚合合成，例如聚乙二醇-聚乳酸-聚乙二醇水凝胶^[58]。这些水凝胶可以是均聚物，共聚物或者多聚物。

杂合水凝胶是由天然聚合物和合成聚合物合成而成。Wang 等人在 1999 年合成了一种蛋白质长链-甲基丙烯酸羟丙酯交联的杂合水凝胶^[59]。

(2) 根据组成成分分类

均聚水凝胶(Homopolymer Hydrogel)是由一类单体衍生而来的交联聚合物网络。这类水凝胶的网络结构取决于单体、聚合技术和交联剂的性质。例如聚乙二醇水凝胶就是一种均聚物水凝胶，它适用于药物、生物分子和蛋白质的控释，并广泛应用于药物传递系统^[60]。

共聚物水凝胶(Copolymer Hydrogel)由两种单体制成，其中一种单体是亲水的，使得聚合物网络能够吸水膨胀(Singhal 和 Gupta 2016)。单体之间的连接可以使交

替分布，也可以是随机分布。这些水凝胶是用引发剂和交联剂使两种单体聚合并且交联合成的。它们的网络之间也存在物理交联。例如羧甲基纤维素水凝胶和羧甲基壳聚糖基水凝胶就是共聚物水凝胶^[61]，它们被用于吸附金属离子。

多聚水凝胶(Multipolymer Hydrogel)是由三种或多种单体通过聚合和交联反应制备而成。例如聚（丙烯酸-2-羟基乙基甲基丙烯酸酯）—明胶水凝胶^[62]。

(3) 根据水凝胶的电性分类

根据水凝胶聚合物长链的电荷性质可以将水凝胶分为以下三种^[63]：

中性水凝胶(Neutral/Non-Ionic Hydrogel)的主链或侧基不带电荷。

离子水凝胶(Ionic Hydrogel)可以是带正电荷的，也可以是带负电荷。带正电基团（如胺基、磺基）的阳离子水凝胶在低 pH 值下溶胀增加，而带负电基团（如羧酸基）的阴离子水凝胶在高 pH 值下溶胀增加。

两性水凝胶(Ampholytic Hydrogel)在同一聚合物链上同时带正电荷基团和负电荷基团，整体电性平衡。

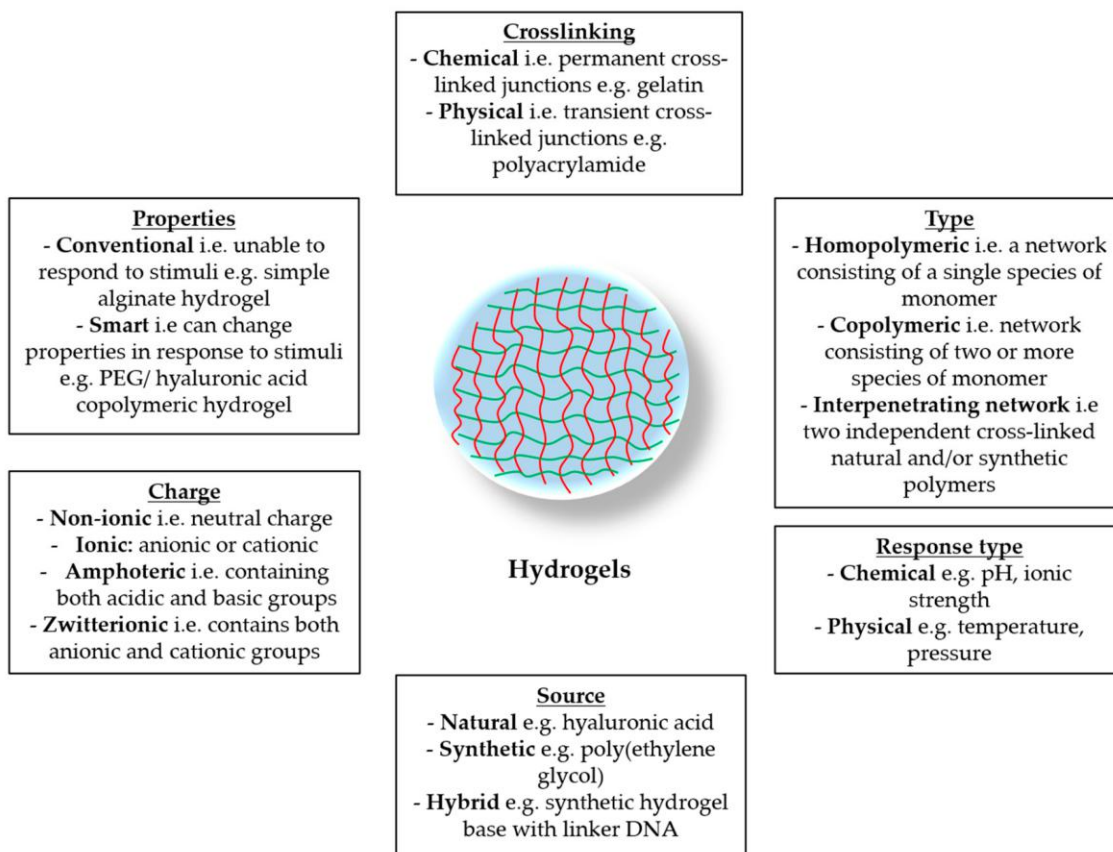


图 2-3 水凝胶的分类

(4) 根据水凝胶聚合物网络的孔径大小分类

根据孔隙率将水凝胶分为三类，即无孔、微孔和超孔^[64]，水凝胶的多孔性使其可以广泛应用于分离技术和药物释放。

(5) 基于水凝胶构型的分类

根据水凝胶的物理结构和化学成分，可以将其分为以下几种^[65]：

非晶型水凝胶(Amorphous/Non-Crystalline Hydrogel)中，聚合物网络包含随机排列的高分子链；而**半晶型水凝胶**(Semi-Crystalline Hydrogel)是一种复杂的非晶相和晶态相的混合物，其特征是密集的有序高分子链区域。

2.4 水凝胶的合成与特性

一个或多个共聚物单体通过共聚和交联可以合成水凝胶。一般来说，引发剂、单体和交联剂是水凝胶的主要合成成分。通过改变引发剂浓度、单体浓度、反应温度、反应容器、反应时间、交联剂与单体的比例等合成参数，可以调节水凝胶的特性。水凝胶高分子网络之间的交联，可以通过以下的方法引入^[66]，包括物理法、化学法、光化学嫁接法、高能辐照法、酶联法，其中化学方法使用最广泛的。大部分实验室合成的水凝胶都是在引发剂和交联剂的存在下，使用化学方法合成的。

化学方法(Chemical Method)是采用各种自由基引发剂，如过硫酸铵(APS)、硝酸铈铵(CAN)、高锰酸钾(KMnO₄)和过硫酸钾(KPS)，将不同乙烯基单体聚合形成高分子长链，然后在交联剂的作用下发生交联，形成聚合物三维网络。如图 2-4 所示：丙烯酸(AAc)和异丙基丙烯酰胺(NIPAAm)是有机单体，2-异丙基硫杂蒽酮(ITX)是引发剂，双季戊四醇五丙烯酸酯(DPEPA)是交联剂。在没有聚合之前，AAC 和 NIPAAm 是单体状态，在 ITX 作用下，单体发生聚合形成长链，在 DPEPA 作用下发成交联，形成聚合物网络^[67]。

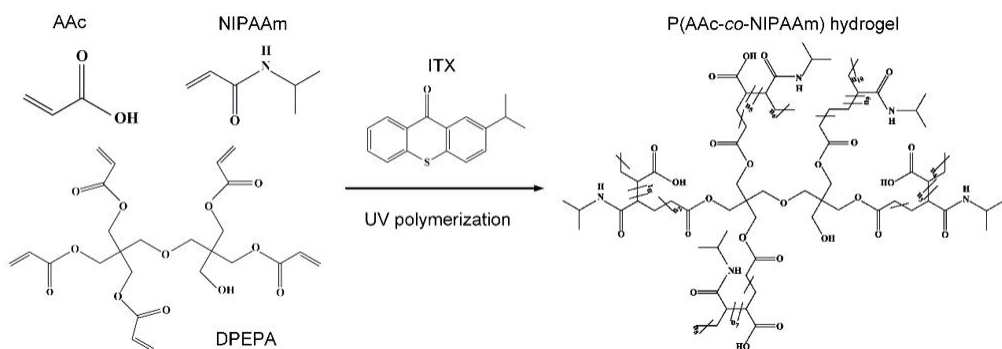


图 2-4 单体在引发剂和交联剂作用下形成聚合物网络^[67]

应用于生物、医学领域的理想的水凝胶应该具有以下特征:高吸水能力、良好

的孔隙率、光稳定性、无毒性、pH 中性、再润湿能力、最低的可溶性含量和残留单体、最高的负载吸收能力(AUL)和高耐久性，水凝胶在溶胀环境下的高稳定性和耐久性使得它在分离过程、药物传递、组织工程和细胞封装等方面非常适用[68]。

2.5 水凝胶的应用

水凝胶具有孔隙率大、溶胀能力强等特点，使其可以广泛应用于分离技术和药物释放。此外，水凝胶在农业、园艺、生物医学和制药等领域有着广泛的应用。

分离技术：水凝胶有良好的溶胀能力和稳定性，是吸附染料、金属杂质、生物分子等水溶性污染物的良好吸附材料。1938 年，Buchholz 和 Graham 首次对聚合物的吸水性能进行了研究[69]。Saraydin 等人在 1994 年研究了马来酸和衣康酸水凝胶的吸附和螯合特性。采用不同比例的马来酸和衣康酸吸附牛血清白蛋白[70]。结果表明，氨基酸含量的增加会增加牛血清白蛋白的吸附量。1997 年，Karadag 等人合成了丙烯酰胺/衣康酸水凝胶吸附剂，用于吸附尼古丁和尼古丁衍生物[71]。某些水凝胶的络合能力使其能够吸附重金属离子[72]。

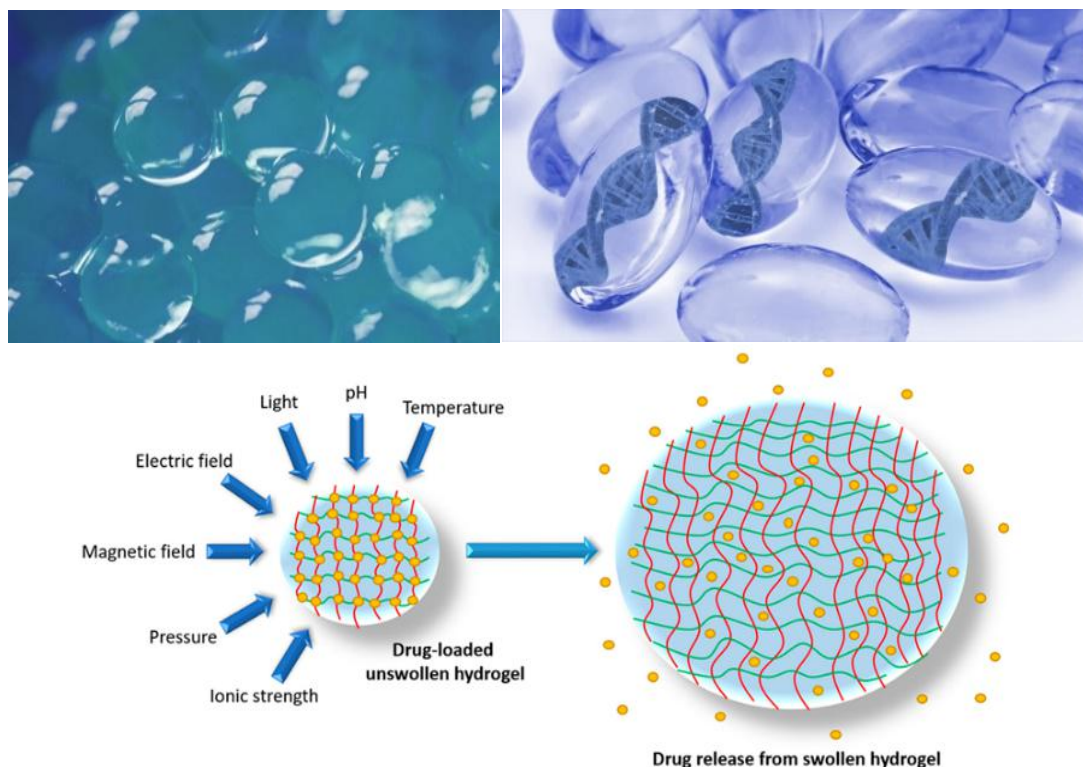


图 2-5 用于医疗的新型 DNA 水凝胶可以响应温度、pH 等刺激释放药物[83]

生物应用：水凝胶是一种柔软且具有弹性的材料，这使得它们在生物医学中

很广泛的应用。水凝胶与一些生物器官和组织在温度、电场、pH 值和酶的变化下的生物学、化学和物理表现非常相似^[76]，因此，水凝胶被广泛用于制作人造器官和组织修复，如肾脏、机器人钳子、人造肌肉、生物传感器、伤口愈合敷料、生物粘合剂、声带替代材料、性器官重建材料和人造皮肤。水凝胶也是制备隐形眼镜的理想选择，因为它们具有良好的机械稳定性、高透氧性和良好的折射率。Wang 等人使用聚己内酯（乙二醇）水凝胶来治疗乳腺癌^[77]。扇贝是水凝胶仿生驱动器的灵感来源^[78, 79]。新型水凝胶致动器在微尺度上可作为微机械臂和微流体系统中的致动器。利用水凝胶在一些刺激下膨胀和收缩特性，可以制作焦距可调的智能光学透镜、还可以制作用于控制液体流动的智能阀门和可切换颜色的材料。

医药学应用：水凝胶具有良好的溶胀性和多孔结构，可以改善一些药物的释放和吸收，在药物传递方面应用已久。常用的给药途径包括通过粘膜、微创口注入和直接服用，水凝胶在控制药物释放速率中起到了重要的作用；水凝胶可以增加药物的溶解度，限制了药物的降解和毒性。以壳聚糖为基础的水凝胶具有良好的生物相容性、低毒性以及可以被人体内的酶降解性，特别适合于药物的传递。由于 pH 敏感性，它们还允许靶向释放。得益于壳聚糖潜在的抗菌和抗氧特性，它们也是最有前途的癌症治疗候选材料之一^[83]。另一个应用是在伤口敷料中。理想的伤口敷料应该吸收多余的渗出物和毒素，并提供一个物理屏障，以保护外部感染源，同时保持伤口湿润^[81]。水凝胶敷料目前有两种形式：凝胶片和局部应用的非晶凝胶。水凝胶敷料可以防止伤口干燥，这有助于肉芽形成、上皮化和自溶清创，使伤口愈合更快^[82]。

水凝胶的另一个医学应用是组织工程。由于器官捐献者和可用器官的非常有限，开发水凝胶用于制作人造器官是非常有意义的。对于严重烧伤或皮肤癌患者，水凝胶也是人造皮肤和皮肤移植的优良材料，水凝胶可以促进新血管和皮肤的形成，并可以通过向细胞发送信号来支持细胞生长和功能。

2.6 水凝胶在力学方面当前研究的主要方向

水凝胶已经广泛应用于生物医学、柔性电子、传感器和仿生驱动器等各个领域。传统水凝胶机械强度通常较低而且容易发生永久性的破坏，这大大限制了水凝胶在力学工程上的应用。促进水凝胶在工程领域的应用和发展需要改进水凝胶的物理化学性质，设计动态调节机制以及更复杂的水凝胶结构。

2.6.1 提高水凝胶的机械性能

常规的水凝胶通常比较柔软易碎，可拉伸性较差（应变小于 100%），并且缺乏断裂韧性。需要承重的人造器官（如软骨组织）、软机器人技术和可穿戴设备的应用等引起了人们对开发一种更加坚韧，可拉伸性更强，力学性能更加优异的

水凝胶的兴趣。

双网络水凝胶作为一种既具有高含水量（约 90 %）又具有高机械强度和韧性的新型材料受到了人们的广泛关注。双网络水凝胶的特征在于由两种力学性质互补的聚合物组分组成：第一层网络是大量交联的聚电解质构成的刚性骨架组成，第二层网络由交联性差的中性聚合物（延性物质）构成，例如（PAMPS-PAAm 水凝胶）。通过两种网络优势互补以实现高的机械强度^[125]，这类双网络水凝胶的拉伸比可以高达 20 倍，断裂能能达到 9000 J/m^2 。在缠结的聚合物网络中引入结晶是获得超强超韧高可拉伸性能水凝胶的另一种方法^[89]。

另一种高强度的水凝胶使用天然蛋白质的衍生物来合成的，这些水凝胶具有非常出色的弹性。例如，弹性蛋白广泛地分布于血管、心脏、膀胱和皮肤组织中，是用于维持组织弹性和完整性的重要成分。通过重组弹性蛋白以及使用酶改变高分子之间的交联，对其进行工程改造^[90]。这些工程化的弹性蛋白基水凝胶通常使用五肽重复序列，具有非常出色的弹性，它们可以拉伸到原始长度的 400%。然而，这些基于蛋白质的凝胶成本非常高，这限制了它们的大范围应用。

纳米材料（例如无机纳米颗粒，碳纳米管和石墨烯）由于具有高度可调节的表面特性和优异的力学性能，因此在合成的水凝胶中添加这些纳米材料，使其与聚合物链相互作用，通过晶体交联^[91]，能显著增强水凝胶整体的力学性能。使用纳米材料作为交联剂可以减少纯聚合物网络中高密度交联的限制。例如，通过使用无机纳米颗粒作为锚点，通过交联将聚合物链连接，得到的 PNIPAAm 水凝胶网络可以拉伸到原来长度的 1400%^[92]。这些添加材料还可以提供额外的功能，如增强导电性或促进组织再生。

在生物医学和柔性电子方面，水凝胶与固体材料（例如金属，玻璃，陶瓷和硅树脂）之间的牢固粘接也是当前的研究热点。这两种材料之间的杨氏模量通常有巨大的差异，在它们之间实现可靠的粘接非常具有挑战性。采用一种由缠绕的聚合物网络制备的复合水凝胶，将聚合物长链通过锚定化学键进一步附着于固体表面^[93]，在剥离过程中，锚定聚合物链的断裂阻力产生粘附，而能量则通过离子交联海藻酸盐的可逆的物理螯合而耗散。这种结合断裂能可以达到 1000 J/m^2 以上，这可以与在人类肌腱和骨骼之间发现的最坚韧的结合（ 800 J/m^2 ）相媲美。除了在生物医学上的潜在应用外，优化水凝胶与其他类型材料（如弹性体）的结合有助于进一步加速柔性电子领域的发展。

2.6.2 水凝胶的断裂和自愈

一些水凝胶自愈能力，使其可以广泛应用在生物医学、外部涂层和柔性电子产品中。这些特性通常依赖于在水凝胶受到外力或破坏后，可以在富水环境中重

建分子间的相互作用。这种自愈能力是通过在水凝胶聚合物网络中加入可逆的物理交联来实现的,包括静电相互作用、氢键和疏水相互作用等。

除了具有交联剂的功能外,纳米颗粒还可以与聚合物链进行物理混合,形成剪切稀化的水凝胶。在水中,由于纳米颗粒和聚合物链之间的强静电相互作用,会形成剪切稀化水凝胶。该水凝胶在剪切应力的作用下其粘度会暂时降低,从而具有可注入性。同时由于静电的相互作用,在低剪切应力时恢复到原来的粘度^[94]。该凝胶可以扩展到使用许多带电的纳米粒子和聚合物链来生产可注射的药剂^[95]。这些剪切稀化水凝胶在止血敷料、治疗血栓、组织工程、药物和基因传递等方面都有应用价值。

在含有静电交联的物理水凝胶中,强离子键作为半永久性的交联来维持水凝胶的形状,而较弱的物理交联通过键可逆的断裂和形成过程来消耗能量,从而提高水凝胶的韧性。通过将氨基或羧基基团等加入到水凝胶网络的聚合物骨架中来增加聚合物网络中氢键的数量,提高其自愈性能^[96],这种水凝胶的自愈能力会受pH变化的影响。

2.6.3 水凝胶致动器

世界上存在的物质,无论是自然存在的还是合成的,都不是完全静态的。在设计应用于生物方面的水凝胶时,我们通常希望水凝胶能够随着时间的推移而发生变化。在过去的十年中,人们设计了各种各样的方法来将动态生理信号整合到水凝胶平台中,从而精确地调节其与细胞组织的相互作用^[97]。

水凝胶的动态调节为研究和实现软体机器人提供了可能性。通过光催化将特定的生物信号添加到基质中,可以使细胞对特定的信号作出反应。这些信号可能包括生长因子、蛋白质、激素和一些细胞粘附肽。

通过对构成水凝胶的聚合物进行仿生设计,它们还可以在特定的刺激下发生变形,这些刺激包括湿度、温度、pH、离子强度、磁场的变化或光刺激。这些水凝胶的结构通常需要异构或多重设计,使其包含不同响应程度的材料,能进行动态的形状变化。例如,由两种具有独特溶胀能力的聚合物(如聚乙二醇和海藻酸盐)形成的双层水凝胶在吸水时表现出定向弯曲^[98]。另一个例子是热敏材料(如PNIPAAm)和热惰性水凝胶材料复合的结构,当温度发生变化时,会产生相应变形^[99]。此外,在水凝胶中加入pH敏感的凝胶会导致结构在pH变化时的发生溶胀或脱水。上述模型依赖于构成水凝胶的聚合物的固有特性,但将磁性纳米颗粒和碳纳米材料等功能材料添加到水凝胶基质中,可以实现结构在磁场的外部刺激或局部光照加热条件下的形变驱动。

早期的生物致动器主要以弹性体材料为主,而水凝胶材料由于具有较好的生

物相容性，近年来受到越来越多的关注。为此，通常采用肌细胞(骨骼肌细胞或心肌细胞)和水凝胶的组合来实现生物致动功能。例如，将碳纳米管-明胶甲基丙烯酸酰胺水凝胶加工成厘米尺度的柔性基质，让心肌细胞自发地、同步地在其表面上跳动，使这些生物执行器表现出有节奏的收缩或伸展和游泳行为^[100]。这些生物致动器使用的水凝胶必须具有生物相容性和机械稳定性。

2.6.4 搭建复杂的结构

生物组织是极其复杂的，人体中几乎所有的器官都包含重复的构建单元，这些单元组装成独特的腔室和界面（例如，心脏中的心肌、心内膜和心包以及构成肝脏的小叶），大多数器官内都有复杂曲折的血管系统，提供氧气和营养物质并运输生物活性分子（如生长因子和激素）。

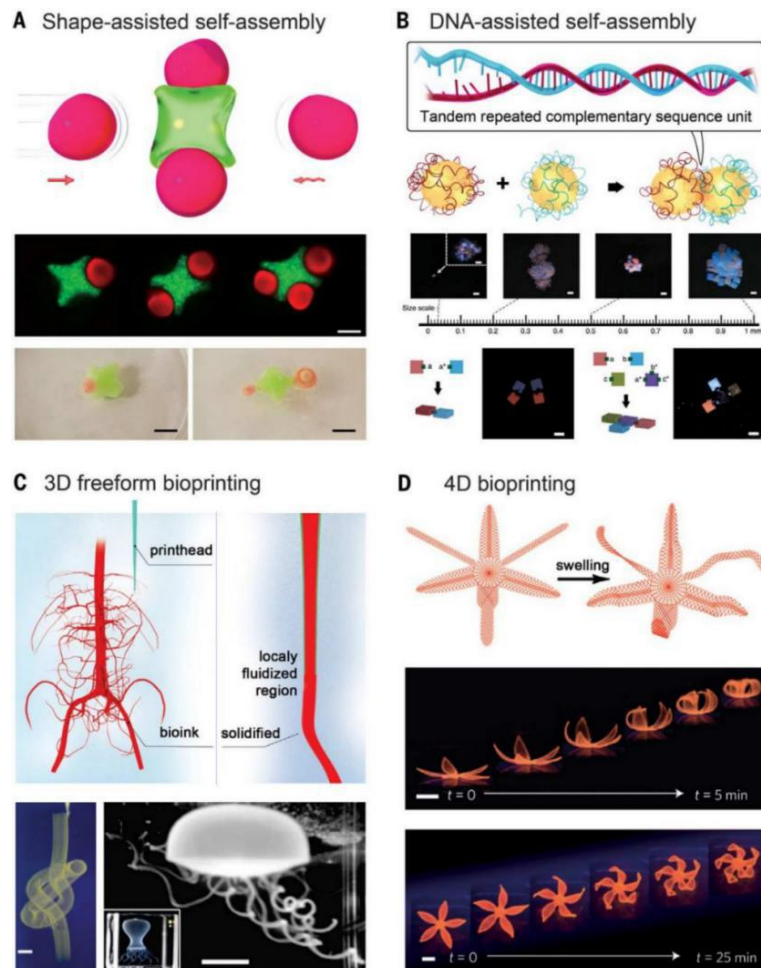


图 2-6 复杂结构水凝胶设计^[30]

这种复杂性还包括血管的结构，血管本身由内膜、中膜和外膜的分层结构。

因此，设计一个仿生器官需要通过合理的设计来重建其固有的特性和生理机能，从而保证其成分、结构和功能的准确性。

由水凝胶制成的软体致动器和电子设备也同样需要类似的结构控制。微工程学(Microengineering)提供了一种通用的方法，通过程序自动组装，通过模块化设计出结构和功能复杂的水凝胶结构。将多种类型的水凝胶模块单元单独制造，利用形状互补、外部能量、表面张力和液-液界面的影响下使不同的模块结合在一起定向组装成块状（图 2-6A）^[101]。利用 DNA 合成的原理设计一种“DNA 凝胶”，结合 DNA 的分子可编程性和微凝胶的形状可控性，可以实现多尺度水凝胶复杂结构的多重组装（图 2-6B）^[102]。近年来，大量的研究人员将 3D 打印技术应用于搭建复杂的水凝胶结构，大大地促进了水凝胶在工业和医学上的应用^[103]。

2.7 本章小结

在过去的十几年中，大量的研究人员致力于提高水凝胶性能，使其在生物医学、柔性电子、传感器和致动器等众多领域的应用机会大大增加。先进的生物制造技术进一步提高了我们构建复杂水凝胶结构的能力，搭建跨尺度的分层组装水凝胶结构。然而，一些挑战也依然存在。水凝胶的临床应用仍需要严格的测试，维持或改善水凝胶在各种介质中的力学性能目前还是一个难题。溶胀是水凝胶的一种常见现象，溶胀常常会导致水凝胶力学性能降低。为了解决水凝胶网络膨胀引起的机械不稳定性，研究水凝胶的溶胀与力学性能之间的关系对于水凝胶的应用非常重要^[104]。

将水凝胶制造与先进的生物制造技术相结合的应用前景也十分广阔。例如，当高可拉伸性的水凝胶与生物打印相结合，可以得到任意形状、图案和结构的弹性基底^[105]。也有研究人员尝试结合数学建模、4D 打印等技术来制造复杂的水凝胶结构。随着跨学科合作、多学科融合发展，将不同领域的先进技术结合起来，致力于提高水凝胶的力学性能、设计制造更实用的结构，引进一些新的功能，并结合其工程实际应用，推动水凝胶的进一步创新发展。

第 3 章 水凝胶理论模型

3.1 化学凝胶溶胀理论

3.1.1 水凝胶的溶胀与大变形理论

当高分子交联形成的弹性网络与小分子溶剂（水，乙醇等）接触时，小分子可以迁移到网络中，形成凝胶。凝胶是一种充满空间的凝聚态物质，小分子很容易发生迁移改变其所在的位置，交联可以防止长聚合物在溶剂中溶解，并且随着小分子的进出，凝胶的体积也会可逆地膨胀和收缩。

下面简单介绍一下聚合物凝胶中质量输运与大变形耦合的场理论。为了弄清这个理论，必须把力学、热力学、动力学原理以及实验联系起来。水凝胶的变形存在两种模式。第一种模式由高分子的局部重排导致的，这种变形模式凝胶只改变形状不改变体积，并且这种模式发生在一个与样本大小无关的时间尺度上。第二种模式是由于溶剂小分子的长距离迁移，它引起凝胶形状和体积的改变。这种变形模式发生在一个时间尺度上，当样本比较大时，这个时间尺度也会变长。这两种模式都能引起大变形。对于溶剂分子的运动，局部重排比长距离迁移快得多。因此在该理论中，假设局部重排是瞬时的，并将小分子的长距离迁移建模为一个依赖于时间的过程^[8]。

聚合物凝胶溶胀或者退溶胀的过程可以抽象成如图 3-1 所示：将所有的外力抽象为一个重力块，小分子的迁移机制可以抽象为一个泵，它们分别对应两种不同的做功方式。重力块施加一个力 P ，下降距离 dl ，做功为 Pdl 。泵维持小分子的化学势 μ ，向凝胶中注入一定数量的 dM ，做功为 μdM 。

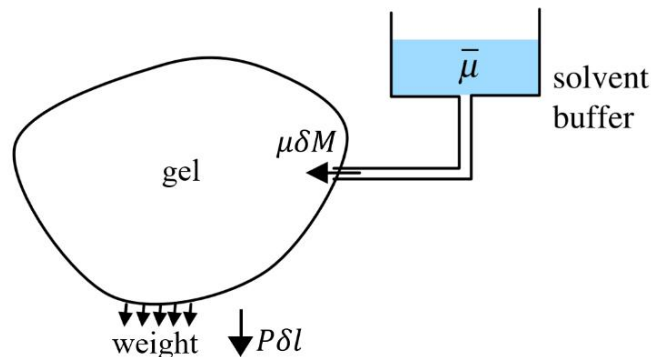


图 3-1 抽象凝胶的受力变形和溶胀退溶胀过程^[107]

如图 3-1 所示，在外力的作用下凝胶发生变形，变形梯度定义 \mathbf{F} 为：

$$F_{iK} = \frac{\partial x_i(\mathbf{X}, t)}{\partial X_K} \quad (3-1)$$

其中 \mathbf{X} 和 \mathbf{x} 分别为参考构型和当前构型中同一物质点的坐标。

外力平衡公式为^[8]：

$$\int s_{iK} \frac{\partial \xi_i}{\partial X_K} dV = \int B_i \xi_i dV + \int T_i \xi_i dA \quad (3-2)$$

式中 s —— 名义应力；

\mathbf{T} —— 面力；

\mathbf{B} —— 体力。

除特别说明之处外，本论文中所有公式的指标按照爱因斯坦求和约定，同一个指标出现两次表示在空间维度上的求和。

由散度定理可得：

$$\int s_{iK} \frac{\partial \xi_i}{\partial X_K} dV = \int (s_{iK}^- - s_{iK}^+) N_K \xi_i dA - \int \frac{\partial s_{iK}}{\partial X_K} \xi_i dV \quad (3-3)$$

$$\frac{\partial s_{iK}(\mathbf{X}, t)}{\partial X_K} + B_i(\mathbf{X}, t) = 0 \quad (3-4)$$

$$(s_{iK}^-(\mathbf{X}, t) - s_{iK}^+(\mathbf{X}, t)) N_K(\mathbf{X}, t) = T_i(\mathbf{X}, t) \quad (3-5)$$

而外部化学势 μ 做功可以被简化成一个泵，将小分子泵入到凝胶内。在当前时刻，单位时间进入单位体积的小分子量为 $r(\mathbf{X}, t) dV(\mathbf{X})$ ，通过单位面积进入的小分子量为 $i(\mathbf{X}, t) dA(\mathbf{X})$ ；而当前时刻单位体积内的小分子量为 $C(\mathbf{X}, t) dV(\mathbf{X})$ ，穿过单位面积的小分子量为 $J_K(\mathbf{X}, t) N_K(\mathbf{X}, t) dA(\mathbf{X})$ 。由小分子的数量守恒可得^[8]：

$$\frac{\partial C(\mathbf{X}, t)}{\partial t} + \frac{\partial J_K(\mathbf{X}, t)}{\partial X_K} = r(\mathbf{X}, t) \quad (3-6)$$

式中 C —— 小分子的浓度；

\mathbf{J} —— 单位面积小分子的通量；

r —— 单位体积小分子的增量。

由散度定理可得：

$$(J_K^+(\mathbf{X}, t) - J_K^-(\mathbf{X}, t)) N_K(\mathbf{X}, t) = i(\mathbf{X}, t) \quad (3-7)$$

$$\int \frac{\partial C}{\partial t} \zeta dV = \int J_K \frac{\partial \zeta}{\partial X_K} dV + \int r \zeta dV + \int i \zeta dA \quad (3-8)$$

由上述推论可得外力的做功功率为 $\int B_i \delta x_i / \delta t dV + \int T_i \delta x_i / \delta t dA$ ，化学势的做功功率为 $\int \mu r dV + \int \mu i dA$ 。凝胶的自由能密度 W 是形变梯度和小分子浓度的函数 $W(\mathbf{F}, C)$ ，并且有：

$$\delta W = \frac{\partial W(\mathbf{F}, C)}{\partial F_{iK}} \delta F_{iK} + \frac{\partial W(\mathbf{F}, C)}{\partial C} \delta C \quad (3-9)$$

则对于图 3-1 中由凝胶，重物 and 泵组成的热力学系统，该系统的自由能 G 的变化率为^[8]：

$$\frac{\delta G}{\delta t} = \int \frac{\delta W}{\delta t} dV - \int B_i \frac{\delta x_i}{\delta t} dV - \int T_i \frac{\delta x_i}{\delta t} dA - \int \mu r dV - \int \mu i dA \quad (3-10)$$

将公式(3-4)、(3-8)、(3-9)代入公式(3-10)可得：

$$\begin{aligned} \frac{\delta G}{\delta t} = & \int \left(\frac{\partial W}{\partial F_{iK}} - s_{iK} \right) \frac{\delta F_{iK}}{\delta t} dV + \int \left(\frac{\partial W}{\partial C} - \mu \right) r dV \\ & + \int \left(\frac{\partial W}{\partial C} - \mu \right) i dA + \int J_K \frac{\partial}{\partial X_K} \left(\frac{\partial W}{\partial C} \right) dV \end{aligned} \quad (3-11)$$

由热力学第二定理可知，对于这样一个孤立系统：

$$\frac{\delta G}{\delta t} \leq 0 \quad (3-12)$$

当我们将在外力作用下高分子网络的重新排布看作是瞬态过程时，公式(3-11)右边第一项可以消去，即：

$$s_{iK} = \frac{\partial W(\mathbf{F}, C)}{\partial F_{iK}} \quad (3-13)$$

假设泵中的每个小分子和凝胶中的小分子之间存在局部平衡，公式(3-11)右边第二项可以消去，即：

$$\mu = \frac{\partial W(\mathbf{F}, C)}{\partial C} \quad (3-14)$$

我们将把小分子的长距离迁移建模为一个时变瞬态过程。为了确保公式(3-11)对任何通量都成立，我们要求公式(3-11)中的第四个被积函数是负定的。一种常见的方法是采用线性动力学定律^[8]：

$$J_K = -M_{KL} \frac{\partial \mu(\mathbf{X}, t)}{\partial X_L} \quad (3-15)$$

式中 M_{KL} —— 迁移率张量(它是一个对称张量)。

聚合物凝胶的自由能来自两个部分：(1)聚合物网络的弹性能；(2)聚合物与小

分子的混合能。根据 Flory 的理论^[106]，假设凝胶的自由能的形式为：

$$\begin{aligned} W(\mathbf{F}, C) &= W_s(\mathbf{F}) + W_m(C) \\ W_s(\mathbf{F}) &= \frac{1}{2} NkT (F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F})) \\ W_m(C) &= -\frac{kT}{\Omega} \left[\Omega C \ln \left(1 + \frac{1}{\Omega C} \right) + \frac{\chi}{1 + \Omega C} \right] \end{aligned} \quad (3-16)$$

分别使用 kT 和 kT/Ω 将化学势和应力归一化。每个分子体积的约为 $\Omega = 10^{-28} \text{ m}^3$ 。常温下， $kT = 4 \times 10^{-21} \text{ J}$ ， $kT/v = 4 \times 10^{-21} \text{ Pa}$ 。自由能函数 (3-16) 引入两个无量纲参数： $Nv = N\Omega$ 和 χ 。在没有小分子的情况下，交联聚合物在小变形下的剪切模量可以用 NkT 来表示， $NkT = 10^4 - 10^7 \text{ N/m}^2$ ，则无量纲弹性模量 Nv 的范围大概为 $10^{-4} - 10^{-1}$ ， χ 是表征混合焓的无量纲参数，范围大概在 $0 \sim 1.2$ 之间。

分子的不可压缩性可以表示为：

$$1 + \Omega C = \det \mathbf{F} \quad (3-17)$$

其中 Ω 是单个小分子的体积，不可压缩性也可以使用拉氏乘子 Π 来表示，并且可以推导出：

$$\left\{ \begin{aligned} s_{iK} &= \frac{\partial W(\mathbf{F}, C)}{\partial F_{iK}} \Rightarrow s_{iK} = \frac{\partial W(\mathbf{F}, C)}{\partial F_{iK}} - \Pi H_{iK} \det \mathbf{F} \\ \mu &= \frac{\partial W(\mathbf{F}, C)}{\partial C} \Rightarrow \mu = \frac{\partial W(\mathbf{F}, C)}{\partial C} + \Pi v \\ \sigma_{ij} &= \frac{\partial W(\mathbf{F}, C)}{\partial F_{iK}} \frac{F_{jK}}{\det \mathbf{F}} - \Pi \delta_{ij} \end{aligned} \right. \quad (3-18)$$

三个主方向的名义应力为：

$$\left\{ \begin{aligned} s_1 &= NkT (\lambda_1 - \lambda_1^{-1}) - \Pi \lambda_2 \lambda_3 \\ s_2 &= NkT (\lambda_2 - \lambda_2^{-1}) - \Pi \lambda_3 \lambda_1 \\ s_3 &= NkT (\lambda_3 - \lambda_3^{-1}) - \Pi \lambda_1 \lambda_2 \end{aligned} \right. \quad (3-19)$$

化学势为：

$$\mu = kT \left[\ln \frac{\Omega C}{1 + \Omega C} + \frac{1}{1 + \Omega C} + \frac{\chi}{(1 + \Omega C)^2} \right] + \Pi \Omega \quad (3-20)$$

上述的凝胶的溶胀和大变形耦合理论可以总结为两个控制方程：(1) 溶剂分子守恒；(2) 力的平衡方程。

(1) 溶剂分子守恒：

$$\dot{C} + \frac{\partial J_K}{\partial X_K} = 0 \quad (3-21)$$

$$\frac{\partial J_L}{\partial X_L} + \frac{\det \mathbf{F}}{\Omega} H_{Ki} \frac{\partial v_i}{\partial X_K} = 0 \quad (3-22)$$

$$J_K = -\frac{D}{kT} \frac{\det \mathbf{F} - 1}{\Omega \det \mathbf{F}} H_{Ki} H_{Li} \frac{\partial \mu}{\partial X_L} \quad (3-23)$$

(2) 力的平衡方程:

$$\frac{\partial s_{iK}}{\partial X_K} = 0 \rightarrow -\delta(W_s + W_m) + \mu \delta C = 0 \quad (3-24)$$

两个控制方程之间通过分子的不可压缩性相关联:

$$1 + \Omega C = \det \mathbf{F} \quad (3-25)$$

3.1.2 水凝胶的溶胀控制方程推导

根据章节 3.1.1 中提及的理论, 假设水凝胶能量的耗散仅通过溶剂小分子的迁移和粘性耗散^[107], 则:

$$\int \dot{U} dV - \int t_i v_i dA - \int \mu l dA + \int \dot{W}_1 dV + \int \dot{W}_2 dV = 0 \quad (3-26)$$

式中 $\dot{W}_1 = R_{KL}^s J_K J_L$ —— 溶剂小分子迁移导致的能量耗散;

$\dot{W}_2 = R_{iKjL}^v \frac{\partial v_i}{\partial X_K} \frac{\partial v_j}{\partial X_L}$ —— 粘性耗散;

$$R_{KL}^s = \frac{F_{iK} F_{iL}}{m \det \mathbf{F}};$$

$$R_{iKjL}^v = \eta \det \mathbf{F} \left[\frac{1}{2} (H_{Km} H_{Lm} \delta_{ij} + H_{Li} H_{Kj}) - \frac{1}{3} H_{Ki} H_{Lj} \right].$$

$$\int \frac{\partial U}{\partial F_{iK}} \frac{\partial v_i}{\partial X_K} dV - \int t_i v_i dA - \int \mu l dA + \int R_{KL}^s J_K J_L dV + \int R_{iKjL}^v \frac{\partial v_i}{\partial X_K} \frac{\partial v_j}{\partial X_L} dV = 0 \quad (3-27)$$

公式(3-27)与以下最小化能量泛函的变分等效:

$$\Pi[\mathbf{v}, \mathbf{J}] = \int \dot{U} dV - \int t_i v_i dA - \int \mu l dA + \int \frac{1}{2} R_{KL}^s J_K J_L dV + \int \frac{1}{2} R_{iKjL}^v \frac{\partial v_i}{\partial X_K} \frac{\partial v_j}{\partial X_L} dV \quad (3-28)$$

$$\delta \Pi[\mathbf{v}, \mathbf{J}] = \int \delta \dot{U} dV - \int t_i \delta v_i dA - \int \mu \delta l dA + \delta \int \frac{1}{2} R_{KL}^s J_K J_L dV + \delta \int \frac{1}{2} R_{iKjL}^v \frac{\partial v_i}{\partial X_K} \frac{\partial v_j}{\partial X_L} dV$$

$$= \int \frac{\partial U}{\partial F_{iK}} \frac{\partial \delta v_i}{\partial X_K} dV - \int t_i \delta v_i dA - \int \mu \delta l dA + \int R_{KL}^s J_K \delta J_L dV + \int R_{iKjL}^v \frac{\partial v_i}{\partial X_K} \frac{\partial \delta v_j}{\partial X_L} dV = 0$$

将溶剂小分子的守恒条件添加到系统的势函数方程中:

$$\begin{aligned}\widehat{\Pi}[\mathbf{v}, \mathbf{J}, \mu] = & \int \dot{U} dV - \int t_i v_i dA - \int \mu IdA + \int \frac{1}{2} R_{KL}^s J_K J_L dV \\ & + \frac{1}{2} \int R_{iKjL}^v \frac{\partial v_i}{\partial X_K} \frac{\partial v_j}{\partial X_L} dV - \int \mu \left[\frac{\det \mathbf{F}}{\Omega} H_{Ki} \frac{\partial v_i}{\partial X_K} + \frac{\partial J_L}{\partial X_L} \right] dV\end{aligned}\quad (3-29)$$

$$\begin{aligned}\widehat{\Pi}[\mathbf{v}, \mathbf{J}, \mu] = & \int \frac{\partial U}{\partial F_{iK}} \frac{\partial v_i}{\partial X_K} dV - \int t_i v_i dA - \int \mu IdA + \int \frac{1}{2} R_{KL}^s J_K J_L dV \\ & + \frac{1}{2} \int R_{iKjL}^v \frac{\partial v_i}{\partial X_K} \frac{\partial v_j}{\partial X_L} dV - \int \mu \left[\frac{\det \mathbf{F}}{\Omega} H_{Ki} \frac{\partial v_i}{\partial X_K} + \frac{\partial J_L}{\partial X_L} \right] dV\end{aligned}\quad (3-30)$$

其中自由能项可以展开为:

$$\int \dot{U} dV = \int \frac{\partial U}{\partial F_{iK}} \frac{\partial v_i}{\partial X_K} dV = \int s_{iK} \frac{\partial v_i}{\partial X_K} dV = \int s_{iK} N_K v_i dA - \int \frac{\partial s_{iK}}{\partial X_K} v_i dV \quad (3-31)$$

将上述公式代入势函数方程中可得:

$$\begin{aligned}& - \int \frac{\partial s_{iK}}{\partial X_K} \delta v_i dV + \int (s_{iK} N_K - t_i) \delta v_i dA \\ & + \int (\mu - \bar{\mu}) \delta IdA + \int \left(R_{KL}^s J_K + \frac{\partial \mu}{\partial X_L} \right) \delta J_L dV \\ & - \int \left[\frac{\det \mathbf{F}}{\Omega} H_{Ki} \frac{\partial v_i}{\partial X_K} + \frac{\partial J_L}{\partial X_L} \right] \delta \mu dV = 0\end{aligned}\quad (3-32)$$

其中:

$$s_{iK} = \frac{\partial U}{\partial F_{iK}} + R_{iKjL}^v \frac{\partial v_j}{\partial X_L} - \mu \frac{\det \mathbf{F}}{\Omega} H_{Ki} \quad (3-33)$$

令上述公式(3-32) 对任意的函数 δv 、 $\delta \mathbf{J}$ 、 $\delta \mu$ 都成立, 则有:

$$\begin{aligned}\frac{\partial s_{iK}}{\partial X_K} &= 0 \\ s_{iK} N_K &= t_i \\ \mu &= \bar{\mu}\end{aligned}\quad (3-34)$$

水凝胶的自由能包含弹性能和混合能, 其表达式见公式(3-16):

$$U = W_s + W_m \quad (3-35)$$

$$\begin{aligned}\widehat{\Pi}[\mathbf{v}, \mathbf{J}, \mu] = & \int \dot{U} dV - \int t_i v_i dA - \int \bar{\mu} IdA + \int \dot{W}_1 dV + \int \dot{W}_2 dV \\ & - \int \mu \left[\frac{\det \mathbf{F}}{\Omega} H_{Ki} \frac{\partial v_i}{\partial X_K} + \frac{\partial J_L}{\partial X_L} \right] dV\end{aligned}\quad (3-36)$$

式中 $\dot{W}_1 = \det \mathbf{F} \frac{\dot{J}_i \dot{J}_i}{m} = R_{KL}^s J_K J_L$;

$$\begin{aligned}\dot{W}_2 &= \eta \det \mathbf{F} \left[\frac{1}{2} \frac{\partial v_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{1}{3} \left(\frac{\partial v_k}{\partial x_k} \right)^2 \right] = R_{ikjL}^v \frac{\partial v_i}{\partial X_K} \frac{\partial v_j}{\partial X_L}; \\ \frac{\partial v_i}{\partial x_j} &= H_{Kj} \frac{\partial \hat{v}_i}{\partial \hat{X}_K}; \\ j_i &= \frac{F_{iK} J_K}{\det \mathbf{F}}.\end{aligned}$$

根据上述内容, 水凝胶的控制方程可以选择四种形式:

(1) 以公式(3-36)为控制方程, 给出 U 、 W_1 、 W_2 的具体形式:

$$U(\mathbf{F}, C) = \frac{1}{2} NkT \left[F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F}) \right] - \frac{kT}{\Omega} \left[\Omega C \ln \left(1 + \frac{1}{\Omega C} \right) + \frac{\chi}{1 + \Omega C} \right] \quad (3-37)$$

对上述公式(3-36)进行变分得到控制方程弱形式:

$$\begin{aligned}\delta \hat{\Pi}[\mathbf{v}, \mathbf{J}, \mu] &= \int \left(\frac{\partial U}{\partial F_{iK}} - R_{ikjL}^v \frac{\partial v_j}{\partial X_L} \right) \frac{\partial \delta v_i}{\partial X_K} dV - \int t_i \delta v_i dA - \int \mu \delta I dA \\ &\quad + \int R_{KL}^s J_K \delta J_L dV - \int \left[\frac{\det \mathbf{F}}{\Omega} H_{Ki} \frac{\partial v_i}{\partial X_K} + \frac{\partial J_L}{\partial X_L} \right] \delta \mu dV\end{aligned} \quad (3-38)$$

(2) 以公式(3-32)为控制方程, 给出 s_{iK} 、 J_K 的具体形式:

$$s_{iK} = \frac{\partial U}{\partial F_{iK}} + R_{ikjL}^v \frac{\partial v_j}{\partial X_L} - \mu \frac{\det \mathbf{F}}{\Omega} H_{Ki} \quad (3-39)$$

$$\begin{aligned}s_{iK} &= NkT (F_{iK} - H_{Ki}) + \frac{kT}{\Omega} \left[\ln \left(1 - \frac{1}{\det \mathbf{F}} \right) \det \mathbf{F} + 1 + \frac{\chi}{\det \mathbf{F}} \right] H_{Ki} \\ &\quad + \eta \det \mathbf{F} H_{Lj} \left(\frac{1}{2} H_{Kj} \frac{\partial v_i}{\partial X_L} + \frac{1}{2} H_{Ki} \frac{\partial v_j}{\partial X_L} - \frac{1}{3} H_{Ki} \frac{\partial v_j}{\partial X_L} \right) - \frac{\mu}{\Omega} \det \mathbf{F} H_{Ki}\end{aligned} \quad (3-40)$$

$$J_K = -\frac{D}{kT} \frac{\det \mathbf{F} - 1}{\Omega} H_{Ki} H_{Li} \frac{\partial \mu}{\partial X_L} \quad (3-41)$$

对控制方程(3-32)进行变分得到控制方程弱形式:

$$\begin{aligned}-\int \frac{\partial s_{iK}}{\partial X_K} \delta v_i dV &+ \int (s_{iK} N_K - t_i) \delta v_i dA + \int (\mu - \bar{\mu}) \delta I dA \\ &+ \int \left(R_{KL}^s J_K + \frac{\partial \mu}{\partial X_L} \right) \delta J_L dV - \int \left[\frac{\det \mathbf{F}}{\Omega} H_{Ki} \frac{\partial v_i}{\partial X_K} + \frac{\partial J_L}{\partial X_L} \right] \delta \mu dV = 0\end{aligned} \quad (3-42)$$

(3) 在第二种方式的基础上进行归一化(使用 NkT 归一化所有的应力、压强和模量等, 使用 L_0 归一化所有的长度量, 使用 $L_0/\Omega t_0$ 归一化所有的通量项,

使用 t_0 归一化所有的时间量), 其中:

$$t_0 = \frac{\eta}{NkT}, L_0 = \sqrt{Dt_0} = \sqrt{\frac{D\eta}{NkT}} \quad (3-43)$$

使用上述量归一化可以得到:

$$\begin{aligned} \hat{s}_{iK} = & F_{iK} - H_{Ki} + \frac{1}{N\Omega} \left[\ln \left(1 - \frac{1}{\det \mathbf{F}} \right) \det \mathbf{F} + 1 + \frac{\chi}{\det \mathbf{F}} - \hat{\mu} \det \mathbf{F} \right] H_{Ki} \\ & + \det \mathbf{F} H_{Lj} \left(\frac{1}{2} H_{Kj} \frac{\partial \hat{v}_i}{\partial \hat{X}_L} + \frac{1}{2} H_{Ki} \frac{\partial \hat{v}_j}{\partial \hat{X}_L} - \frac{1}{3} H_{Ki} \frac{\partial \hat{v}_j}{\partial \hat{X}_L} \right) \end{aligned} \quad (3-44)$$

$$J_K = -\frac{D}{kT} \frac{\det \mathbf{F} - 1}{\Omega} H_{Ki} H_{Li} \frac{\partial \mu}{\partial X_L} \quad (3-45)$$

对控制方程(3-32)进行变分:

$$\begin{aligned} & \int \hat{s}_{iK} \frac{\partial \delta \hat{v}_i}{\partial \hat{X}_K} dV - \int \hat{t}_i \delta \hat{v}_i dA - \int \frac{\det \mathbf{F}}{N\Omega} H_{Ki} \frac{\partial \hat{v}_i}{\partial \hat{X}_K} \delta \hat{\mu} dV \\ & - \int \frac{\det \mathbf{F} - 1}{N\Omega} H_{Ki} H_{Li} \frac{\partial \hat{\mu}}{\partial \hat{X}_L} \frac{\partial \delta \hat{\mu}}{\partial \hat{X}_K} dV \\ & + \int \frac{\det \mathbf{F} - 1}{N\Omega} H_{Ki} H_{Li} \frac{\partial \hat{\mu}}{\partial \hat{X}_L} N_K \delta \hat{\mu} dA = 0 \end{aligned} \quad (3-46)$$

(4) 参考章节 3.1.1 中公式(3-21)(3-24), 忽略粘性耗散, 利用力的平衡条件以及溶剂小分子的守恒条件组合成系统的控制方程:

$$\int (-\delta \hat{W} + \mu \delta C) dV + \int \left[-(\lambda_0^3 \det \mathbf{F} - 1) H_{Ki} \frac{\partial \mu}{\partial X_K} \frac{\partial \delta \mu}{\partial X_L} H_{Li} - \frac{\partial C}{\partial t} \delta \mu \right] dV = 0 \quad (3-47)$$

其中水凝胶的亥姆霍兹自由能:

$$\hat{W} = \frac{1}{2} Nv \left[\lambda_0^2 I_1 - 3 - 2 \ln(\lambda_0^3 \det \mathbf{F}) \right] + (\lambda_0^3 \det \mathbf{F} - 1) \left[\ln \left(\frac{\lambda_0^3 \det \mathbf{F} - 1}{\lambda_0^3 \det \mathbf{F}} \right) - 1 \right] \quad (3-48)$$

溶剂浓度的变分:

$$\delta C = \lambda_0^3 \delta(\det \mathbf{F}) = \lambda_0^3 \det \mathbf{F} H_{Ki} \delta F_{iK} \quad (3-49)$$

速度的散度:

$$\frac{\partial v_i}{\partial x_i} = \frac{\partial v_i}{\partial X_K} H_{Ki} \quad (3-50)$$

对时间进行离散化:

$$v_i = \frac{x_i^{n+1} - x_i^n}{\Delta t} \quad (3-51)$$

于是速度的散度为:

$$\frac{\partial v_i}{\partial x_i} = \frac{F_{iK}^{n+1} - F_{iK}^n}{\Delta t} H_{Ki} \quad (3-52)$$

溶剂浓度对时间的偏导：

$$\frac{\partial C}{\partial t} = \lambda_0^3 \det \mathbf{F} \frac{\partial v_i}{\partial x_i} = \lambda_0^3 \det \mathbf{F} \frac{\partial v_i}{\partial X_K} H_{Ki} \quad (3-53)$$

$$\begin{aligned} \dot{W}_{vis} &= \eta \det \mathbf{F} \left[\frac{1}{2} \frac{\partial v_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{1}{3} \left(\frac{\partial v_k}{\partial x_k} \right)^2 \right] \\ &= \eta \det \mathbf{F} \left[\frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} \right)^2 + \frac{1}{2} \frac{\partial v_i}{\partial x_j} \frac{\partial v_j}{\partial x_i} - \frac{1}{3} \left(\frac{\partial v_k}{\partial x_k} \right)^2 \right] \end{aligned} \quad (3-54)$$

将上述公式代入到控制方程中展开可得水凝胶溶胀的控制方程弱形式：

$$\begin{aligned} &\int -\delta \left(\frac{1}{2} Nv \left[\lambda_0^2 I_1 - 3 - 2 \ln(\lambda_0^3 \det \mathbf{F}) \right] + (\lambda_0^3 \det \mathbf{F} - 1) \left[\ln \left(\frac{\lambda_0^3 \det \mathbf{F} - 1}{\lambda_0^3 \det \mathbf{F}} \right) - 1 \right] \right) dV \\ &+ \int \mu \lambda_0^3 \det \mathbf{F} H_{Ki} \delta F_{iK} dV + \int \left[-(\lambda_0^3 \det \mathbf{F} - 1) H_{Ki} \frac{\partial \mu}{\partial X_K} \frac{\partial \delta \mu}{\partial X_L} H_{Li} - \lambda_0^3 \det \mathbf{F} \frac{\partial v_i}{\partial X_K} H_{Ki} \delta \mu \right] dV = 0 \end{aligned} \quad (3-55)$$

3.2 水凝胶含水量对弹性模量的影响

描述聚合物均匀膨胀变形行为的最著名模型是 Flory-Huggins 模型^[101]。该方法基于描述聚合物网络弹性响应的经典高斯链模型，以及通过 Flory-Huggins 溶液理论导出聚合物分子与溶剂分子之间的混合能。尽管 Flory-Huggins 模型被广泛用作解释膨胀聚合物力学性质的理论基础^[9]，但是该理论也存在一些不足之处。

根据 Flory-Huggins 模型，凝胶在脱水的过程中弹性模量会变大同时材料会变得更脆，当凝胶吸水时，弹性模量会变低，并且与高分子体积分数的 1/3 次方成正比，即 $E = E_0 \phi^{1/3}$ 。 E_0 是干凝胶的弹性模量。但是实际上 $E = E_0 \phi^v$ ，指数 v 取决于水凝胶的底层弹性体网络，并且溶胀和退溶胀过程中的 v 不相同。

Li 等人的通过一系列的试验^[9]，得出水凝胶的弹性模量 E 与聚合物体积分数 ϕ 的 v 次幂成正比，并且有：

$$E = E_{ini} \phi^{0.56}, \text{ 溶胀时}$$

$$E = E_{ini} \phi^{2.3}, \text{ 退溶胀时}$$

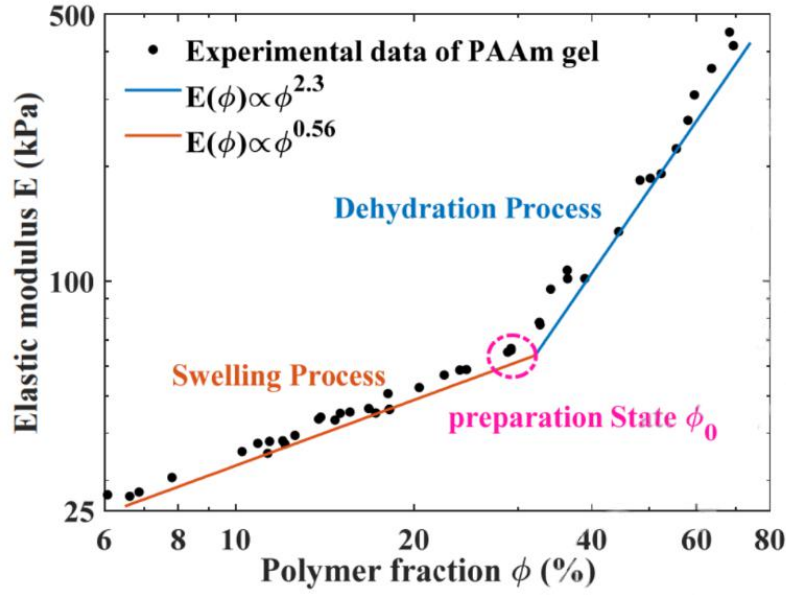


图 3-2 实验测得水凝胶的弹性模量 E 与聚合物体积分数 ϕ 的 ν 次幂成正比，溶胀和退溶胀过程中幂指数 ν 不同^[9]

(1) F-H 模型的标度律 $E = E_0 \phi^{1/3}$ ：

聚合物凝胶的自由能来自两个部分：(1) 聚合物网络的弹性能 W_s ；(2) 聚合物与小分子的混合能 W_m 。

$$W(\mathbf{F}, C) = W_s(\mathbf{F}) + W_m(C) \quad (3-56)$$

在 Flory-Huggins 模型中^[108]：

$$W_s(\mathbf{F}) = \frac{1}{2} NkT (F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F})) \quad (3-57)$$

$$W_m(C) = -\frac{kT}{\Omega} \left[\Omega C \ln \left(1 + \frac{1}{\Omega C} \right) + \frac{\chi}{1 + \Omega C} \right] \quad (3-58)$$

λ_0 是凝胶预溶胀的拉伸比，则预溶胀凝胶的弹性能密度可以表示为：

$$\begin{aligned} W_0(\mathbf{F}) &= \frac{1}{2} \frac{NkT}{\lambda_0^3} (\lambda_0^2 F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F}) - 6 \ln \lambda_0) \\ &= \frac{1}{2} \frac{NkT}{\lambda_0} \left(F_{iK} F_{iK} - \frac{3 + 2 \ln(\det \mathbf{F}) + 6 \ln \lambda_0}{\lambda_0^2} \right) \end{aligned} \quad (3-59)$$

上述变形梯度 \mathbf{F} 都是从预溶胀状态下开始计算的。

假设凝胶的体积不可压，在单轴拉伸的情况下轴向拉伸比 $\lambda_1 = \lambda$ ，则其他两个方向的拉伸比为 $\lambda_2 = \lambda_3 = 1/\lambda^{\frac{1}{2}}$ ，由公式(3-59)可得：

$$s_1 = \frac{\partial W}{\partial \lambda} = \frac{NkT}{\lambda_0} \left(\lambda - \frac{1}{\lambda^2} \right) \quad (3-60)$$

这时凝胶溶胀率以及混合能与其应力无关，可以推导出凝胶的弹性模量为：

$$E = \frac{\partial s}{\partial \lambda} \bigg|_{\lambda=1} = 3 \frac{NkT}{\lambda_0} \quad (3-61)$$

因此可以得到凝胶的弹性模量与高分子体积分数的 $1/3$ 次方成正比 ($\phi = 1/\lambda_0^3$)：

$$E \sim \phi^{1/3} \quad (3-62)$$

(2) 对于制备状态修正的 F-H 模型的标度律：

上述经典的 F-H 模型假设任何一个化学凝胶高分子网络存在一个唯一的参考构型，就是完全退溶胀时干聚合物的状态，并且高分子网络在这一状态下无应力。这一假设对于一般的凝胶制备过程可能是不适用的，在凝胶吸水溶胀或者溶剂发生改变时完全松弛的高分子链的参考构型也会相应发生改变。假设组成高分子长链的单体之间，以及单体与溶剂分子之间没有相互作用，这样的高分子长链称为理想高分子链。

拉伸这种理想的高分子链所需的能量与长链端到端的距离 R 的二次方成正比^[108]：

$$F_{el} \approx kT \frac{(\lambda R_0)^2}{R_{ref}^2} \quad (3-63)$$

其中 R_{ref} 为高分子网络上的长链在不受外力作用下端到端的波动距离，与聚合物体积分数 ϕ 间有如下的比例关系^[108]：

$$R_{ref} \sim \phi^{-\left(\frac{v-1}{2}\right)/(3v-1)} \quad (3-64)$$

与干聚合物相比，长链端到端距离波动为：

$$\lambda_{ref} = R_{ref}/R_0 = \phi^{-\left(\frac{v-1}{2}\right)/(3v-1)} = \lambda_0^{\left(\frac{v-1}{2}\right)\left(\frac{v-1}{3}\right)} \quad (3-65)$$

由这个端到端的距离可以推出凝胶的弹性能密度为：

$$W(\mathbf{F}) = \frac{1}{2} \frac{NkT}{\lambda_0^3} \left(\left(\frac{\lambda_0}{\lambda_{ref}} \right)^2 F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F}) - 6 \ln \left(\frac{\lambda_0}{\lambda_{ref}} \right) \right) \quad (3-66)$$

同样假设凝胶的体积不可压，在单轴拉伸的情况下 $\lambda_1 = \lambda$ ，其他两个方向的拉伸比为 $\lambda_2 = \lambda_3 = 1/\lambda^{1/2}$ ，由公式(3-66)可得：

$$s_1 = \frac{\partial W}{\partial \lambda} = \frac{NkT}{\lambda_0 \lambda_{ref}^2} \left(\lambda - \frac{1}{\lambda^2} \right) \quad (3-67)$$

可以推导出凝胶的弹性模量为：

$$E = \frac{\partial s}{\partial \lambda} \bigg|_{\lambda=1} = 3 \frac{NkT}{\lambda_0 \lambda_{ref}^2} = 3 \frac{NkT}{\lambda_0^{\frac{3-1}{3v-1}}} \quad (3-68)$$

因此可以得到凝胶的弹性模量与高分子体积分数的关系如下（ $v = 0.588$ ）：

$$E \sim \phi^{1 - \frac{1/3}{3v-1}} \sim \phi^{0.56} \quad (3-69)$$

那么实际上退溶胀过程凝胶的弹性模量与高分子体积分数的关系是怎样呢？

(3) 退溶胀过程中的标度律：

根据 Li 等人的试验^[9]，水凝胶从制备状态溶胀和退溶胀时的模量变化的标度律是不同的。为了解释这一现象，他们假设当水凝胶从制备状态开始退溶胀时，每条聚合物链的端到端距离不会因退溶胀而发生改变，换言之，退溶胀时每条聚合物链不会受压。从这个假设出发，退溶胀过程中凝胶的自由能密度是：

$$W(\mathbf{F}) = \frac{1}{2} \frac{NkT}{\lambda_0^3} \left(\left(\frac{1}{\lambda_{ref}} \right)^2 F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F}) + 6 \ln \lambda_{ref} \right) \quad (3-70)$$

同样假设凝胶的体积不可压，在单轴拉伸的情况下 $\lambda_1 = \lambda$ ，则其他两个方向的拉伸比为 $\lambda_2 = \lambda_3 = 1/\lambda^{\frac{1}{2}}$ ，公式(3-73)退化为：

$$s_1 = \frac{\partial W}{\partial \lambda} = \frac{NkT}{\lambda_0^3 \lambda_{ref}^2} \left(\lambda - \frac{1}{\lambda^2} \right) \quad (3-71)$$

可以推导出凝胶的弹性模量为：

$$E = \frac{\partial s}{\partial \lambda} \bigg|_{\lambda=1} = 3 \frac{NkT}{\lambda_0^3 \lambda_{ref}^2} = 3 \frac{NkT}{\lambda_0^{\frac{5-1}{3v-1}}} \quad (3-72)$$

因此可以得到凝胶的弹性模量与高分子体积分数的关系如下：

$$E \sim \phi^{\frac{5-1/3}{3v-1}} \sim \phi^{1.23} \quad (3-73)$$

与 Li 等人的结果不同的是，上述理论导出的标度律仍然与实验^[9]相悖。

为了得到与实验一致的结果，我们进一步假设在水凝胶退溶胀的过程中，随着高分子浓度的增加，高分子链之间的距离会越来越远，这时将会有更多的物理交联(氢键、范德华力、高分子链缠绕等)产生，等效的高分子链数量 N 将会增大。假设标度律：

$$N(\phi) = N_0 \phi^n ; \text{退溶胀时} \quad (3-74)$$

在 $n=1.07$ 时, 可以得到 $E \sim \phi^{2.3}$.

3.3 与 COMSOL 仿真结果对比

根据章节 3.1、3.2 中的理论, 在 COMSOL 中建立相应的水凝胶溶胀和退溶胀模型。设置 COMSOL 模型的关键参数如表 3-1 所示:

表 3-1 COMSOL 模型的关键参数

名称	表达式	值
C0	0.2	0.2
Nv	0.001	0.001
lambda0	$(1+C0)^{(1/3)}$	1.0627
mu0	$\log(C0/(1+C0))+1/(1+C0)+\chi/(1+C0)^2+Nv*(1/\lambda0-1/(1+C0))$	-0.81943
vis	0.01	0.01
v_const	0.588	0.588

其中 C_0 为初始的溶剂份数(假设高分子网络份数为 1), 可得水凝胶此时因为溶胀引起的 $\lambda_0 = (1 + C_{10})^{\frac{1}{3}}$; 水凝胶的无量纲弹性模量 $Nv = 0.001$, 取 $kT/v = 4 \times 10^7$ Pa; vis 为粘度系数, v_const 为标度律指数 $v = 0.588$ 。

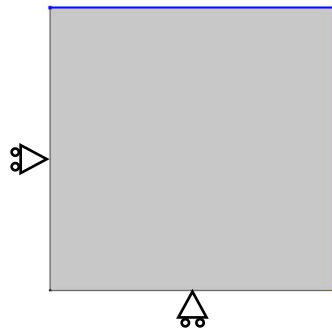


图 3-3 设置溶胀、退溶胀模型的边界条件

设置模型的边界条件和初始化: (1)化学势边界条件: 如上图所示选中的边界, 溶胀过程中其化学势随时间慢慢由 μ_0 线性变化为 0, 其余两条边界为零通量边界条件; 退溶胀过程中其化学势随时间慢慢由 μ_0 线性变化为 $1.5 \times \mu_0$, 其余两条边界为零通量边界条件; (2)位移边界条件: 左边界固定 x 方向的位移, 下边界固定住 y 方向的位移; (3)初始化场: 位移场初始化为 0, 速度场初始化为 0; 凝胶内部初始化学势为 μ_0 , 化学势场的时间导数也为 0。

3.3.1 标度律修正的溶胀模型仿真结果

根据上述表 3-1 中的参数以及图 3-3 中的几何模型和边界条件在 COMSOL 中建立相关的二维水凝胶溶胀模型。

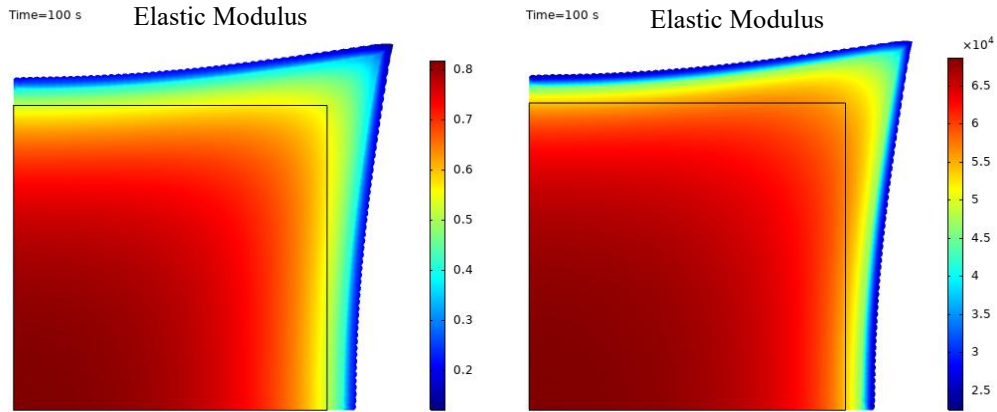
Swelling 模型的弹性能方程选择(3-66):

$$W(\mathbf{F}) = \frac{1}{2} \frac{NkT}{\lambda_0^3} \left(\left(\frac{\lambda_0}{\lambda_{ref}} \right)^2 F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F}) - 6 \ln \left(\frac{\lambda_0}{\lambda_{ref}} \right) \right)$$

对模型进行后处理，输出位移场、化学势场以及弹性模量随时间变化的曲线图：

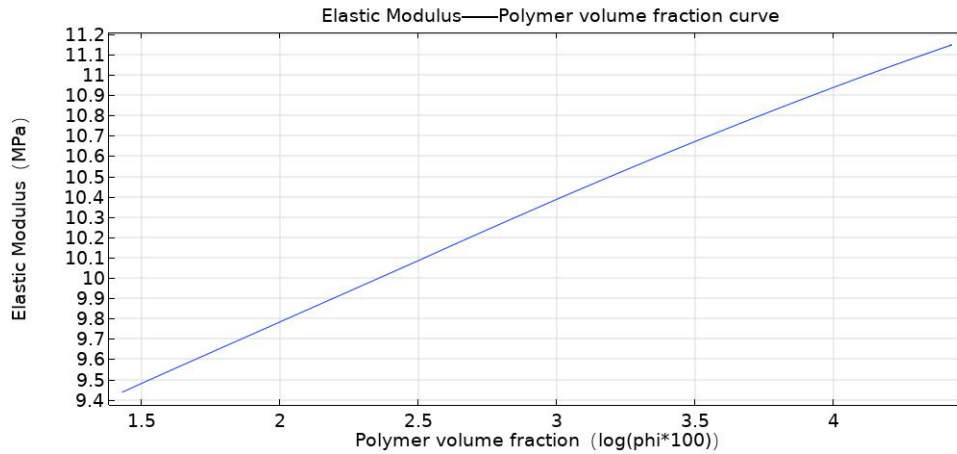
$$E = 3 \frac{NkT}{\lambda \lambda_{ref}^2} = 3 \frac{Nv}{I_1^{1/2} \lambda_{ref}^2} \times 4 \times 10^7$$

模型结果如图 3-4 所示：



a) 高分子链体积份数云图

b) 弹性模量云图



c) 弹性模量与高分子链体积份数关系曲线

图 3-4 标度律模型水凝胶 COMSOL 溶胀模型

3.3.2 标度律修正的退溶胀仿真结果

标度律修正的退溶胀模型的自由能弹性性能方程选择(3-70):

$$W(\mathbf{F}) = \frac{1}{2} \frac{NkT}{\lambda_0^3} \left(\left(\frac{1}{\lambda_{ref}} \right)^2 F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F}) + 6 \ln \lambda_{ref} \right)$$

对模型进行后处理, 输出位移场、化学势场以及弹性模量随时间变化的曲线:

$$E = 3 \frac{NkT}{\lambda \lambda_{ref}^2} = 3 \frac{(\phi/\phi_0)^{1.07} Nv}{I_1^{1/2} \lambda_{ref}^2} \times 4 \times 10^7$$

模型结果如图 3-5 所示:

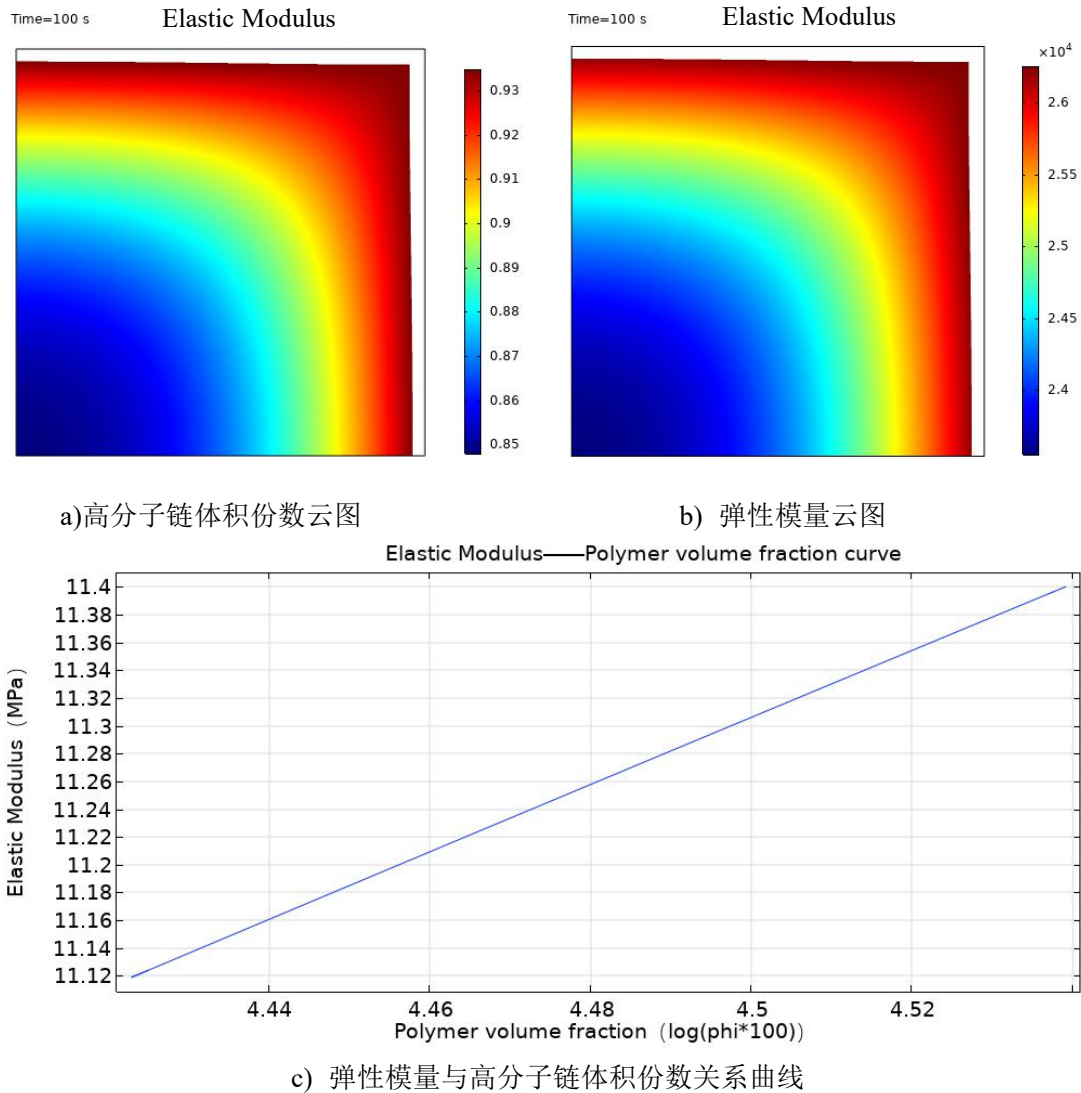


图 3-5 标度律模型水凝胶 COMSOL 退溶胀模型

整理上述结果，绘制溶胀和退溶胀过程中的杨氏模量(对数)与高分子链体积分数(对数)关系曲线，如图 3-6 所示。

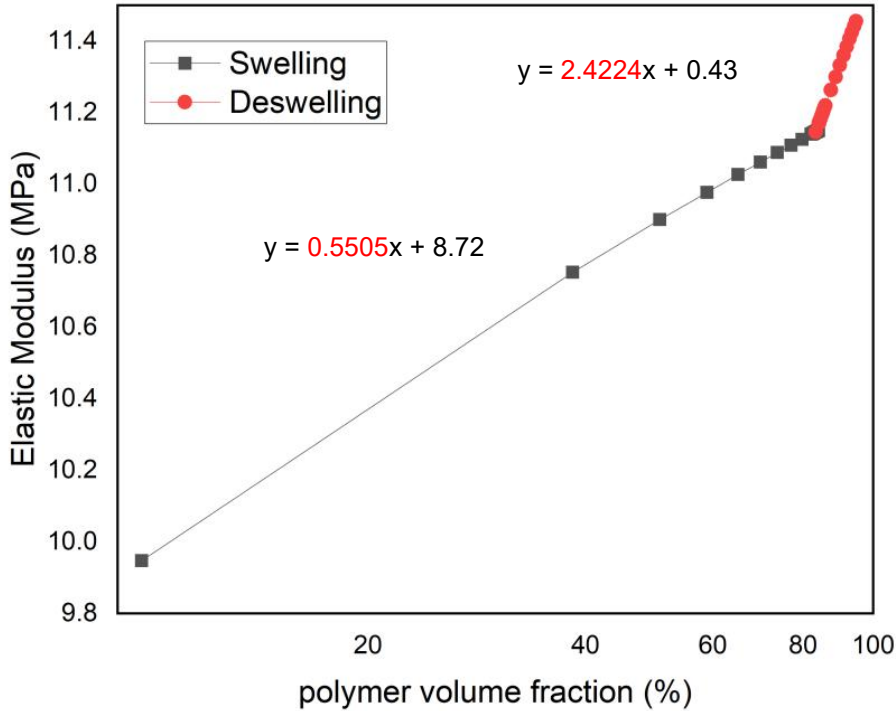


图 3-6 COMSOL 模型测得水凝胶的弹性模量 E 与聚合物体积分数 ϕ 的 ν 次幂成正比，溶胀和退溶胀过程中 ν 不同

上述结果验证了章节 3.2 中修正的标度率理论的合理性。

3.4 化学凝胶溶胀试验

3.4.1 凝胶拉伸试验

最近本课题组开展了一系列试验，使用不同浓度的有机溶液配制组分相同的水凝胶，然后加入适量的水，使不同的水凝胶样本溶胀到其高分子体积分数相同，并进行单轴拉伸试验，研究它们之间力学性能的差异。采用的凝胶是化学交联的聚丙烯酰胺水凝胶。使用四组不同浓度下制备的凝胶，溶胀到相同化学组分，进行单轴拉伸试验，测得其名义应力、弹性模量与拉伸比关系如下图所示：

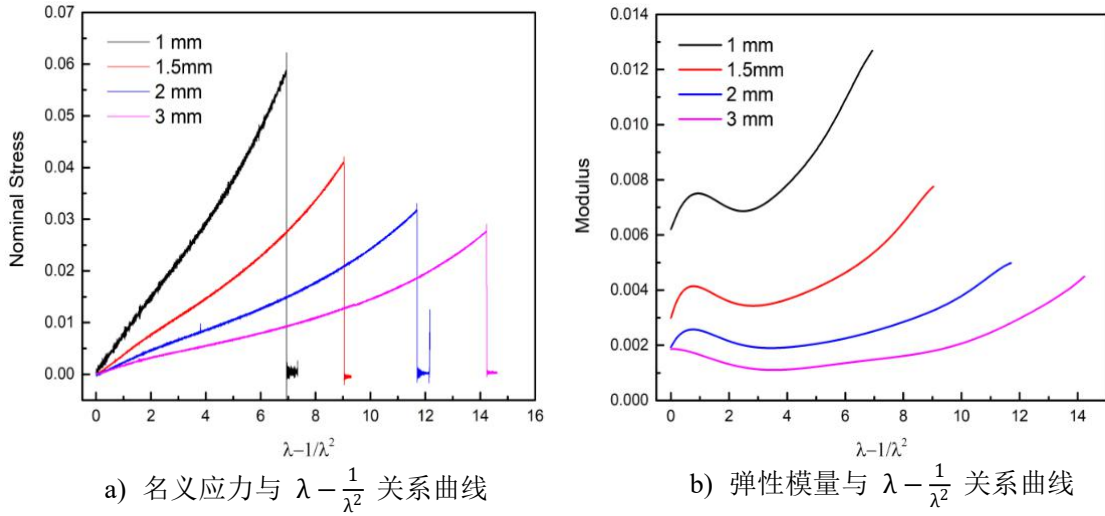


图 3-7 水凝胶单轴拉伸试验结果

3.4.2 理论推导

根据章节 3.2 中的标度律对上述实验现象进行理论解释。不同的样本在初始时刻的高分子链数量 N_0 相同，溶胀导致的预溶胀拉伸比 λ_{0i} 不同，其与原始体积相关，表达式如下：

$$\begin{aligned}\lambda_{0i} &= \lambda_0 i^{1/3}, i=1, 1.5, 2, 3 \\ N &= N_0 \phi^{1.07} \sim N_0 i^{-1.07}\end{aligned}\quad (3-75)$$

将上述关系式代入章节 3.2 中的标度律的溶胀自由能方程中得到：

$$W_{0i}(F) = \frac{1}{2} \frac{N_0 k T}{\lambda_{0i}^3} \left(\left(\frac{\lambda_{0i}}{\lambda_{ref-i}} \right)^2 F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F}) - 6 \ln \left(\frac{\lambda_{0i}}{\lambda_{ref-i}} \right) \right) \quad (3-76)$$

$$W_i = \frac{1}{2} \frac{N k T}{\lambda_{swell}^3} \left(\left(\frac{\lambda_{prep-swell_i}}{\lambda_{ref}} \right)^2 F_{iK} F_{iK} - 3 - 2 \ln(\det \mathbf{F}) - 6 \ln \left(\frac{\lambda_{prep-swell_i}}{\lambda_{ref}} \right) \right) \sim \frac{1}{2} N k T \left(\frac{\left(\frac{3}{i} \right)^{\frac{1}{3}} \lambda}{\lambda_{ref}} \right)^2 \quad (3-77)$$

在单轴拉伸的情况下 $\lambda_1 = \lambda$ ，则其他两个方向的拉伸比为 $\lambda_2 = \lambda_3 = 1/\lambda^{\frac{1}{2}}$ ，名义应力表达式如下：

$$s_i = \frac{NkT\lambda_{prep-swel,i}^2}{2\lambda_{swell}^3\lambda_{ref}^2} \left(\lambda - \frac{1}{\lambda^2} \right) \sim \frac{1}{2} NkT \left(\frac{3}{i} \right)^{\frac{2}{3}} \left(\lambda - \frac{1}{\lambda^2} \right) \sim i^{-1.74} \quad (3-78)$$

在绘制名义应力、弹性模量与 $\lambda - \frac{1}{\lambda^2}$ 的曲线图如下(散点为近似的实验数据):

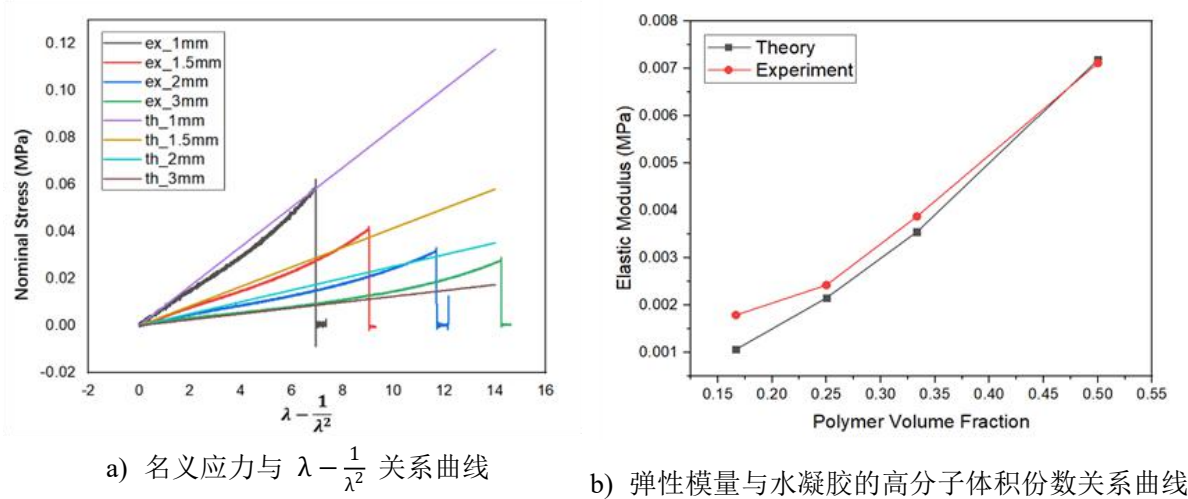


图 3-8 水凝胶单轴拉伸应用标度律结果与试验对比

由上述试验以及理论推导发现标度律可以很好的解释上述试验现象。因为标度律中选用的本构模型是简单的新胡克模型，对 $\lambda - \frac{1}{\lambda^2}$ 偏导得到的不同样本的等效模量为常数，没有拉伸硬化现象，选择使用其他的超弹性体模型可能会与实验结果更加一致。

3.5 本章小结

第三章主要的内容包括：(1) 介绍了化学凝胶溶胀的非平衡态热力学理论模型，并推导了 COMSOL 水凝胶建模及 FEniCS 程序需要使用的化学凝胶溶胀的控制方程；(2) 推导并解释了水凝胶在溶胀退溶胀过程中模量随高分子体积分数变化的理论模型，并将推导的理论模型结合 COMSOL 建立相关模型；(3) 采用上述修正的标度律理论，推导解释了使用不同浓度的有机溶液配制组分相同的水凝胶，溶胀到高分子体积分数相同，力学性能不同的现象。通过上述工作，得出了如下结论：

(1) 化学凝胶溶胀的过程是一个溶剂分子扩散动力学与凝胶宏观力学变形耦合的非平衡态热力学过程；

(2) 根据经典的 Flory-Huggins 模型，水凝胶在溶退溶胀的过程中弹性模量会

随着高分子的体积分数的改变发生变化。本课题从凝胶的自由能方程出发，给出了水凝胶在溶胀退溶胀条件下对应的自由能方程。结果显示，在溶胀过程中，水凝胶的弹性模量与 $\phi^{0.56}$ 成正比，在退溶胀过程中，水凝胶的弹性模量与 $\phi^{2.3}$ 成正比。上述理论与试验结果比较吻合。

(3) 同时本课题应用上述理论在 COMSOL 中建立了相关的水凝胶溶胀退溶胀模型，模型的运行结果也与标度律相吻合；利用上述修正的标度律推导解释了使用不同浓度的有机溶液配制组分相同的水凝胶，使不同凝胶样本吸水溶胀到高分子体积分数相同时，力学性能不同的现象。

第 4 章 基于 FEniCS 的水凝胶力学行为研究

4.1 FEniCS 简介及简单示例

4.1.1 FEniCS 开源有限元平台简单介绍

FEniCS 最初源自芝加哥大学与查尔姆斯理工大学的一个合作项目，是一个用于求解偏微分方程的开源计算平台，其目标是使用户能够快速地将科学模型转换为高效的有限元代码^[6]。关于 FEniCS 的下载，安装和使用等详细信息可以参考项目主页 <https://fenicsproject.org/>。

FEniCS 是由许多核心模块组成的，这些核心模块包括 `dolfin`、`UFL`、`FFC`、`FIAT` 和 `mshr` 等。`dolfin` (dynamic object-oriented library for finite element computation) 是用于有限元计算的面向对象库^[114]，它本质上是 FEniCS 的 C++ 后端，并且配置了 Python 的接口，其主要功能是进行有限元网格划分，函数空间定义，有限元稀疏矩阵组装，网格细化等数据结构及算法实现^[115]。`UFL` (The Unified Form Language) 是一种用于声明有限元离散变分格式 (variational form) 的专用语言^[119]，它建立在 python 语言基础上，所以可以看作是一种推广了的 python 语言^[120]。具体地讲，利用这种语言我们可以以非常贴近数学记法的方式来编写计算程序。`FFC` (The FEniCS Form Compiler)，也就是 FEniCS 中有限元变分型的编译器^[116]，可以把用 UFL 语言写的多线性型编译为更高效的 C++ 代码^[117]。例如，把定义了一个变分问题的双线性型（左端）和线性型（右端）编译成能用于组装对应的线性系统的代码^[118]。`FIAT` (Finite element automatic tabulator) 是有限单元自动生成器^[121]，能够生成一维，二维，三维，三角形，矩形，四面体，六面体等任意阶的有限单元^[122]，`mshr` 就是 FEniCS 的网格生成器。

FEniCS 的特点是在 Python/C++ 语法基础上，进一步针对有限元的数学语言进行了语法抽象，从而可以以一种非常贴近数学语言的方式来编写有限元程序。FEniCS 拥有一套自己的有限元语言的编译器，从而可以自动生成底层代码。由于用户在非常高的抽象层次上进行编程，所以不需要考虑很多底层的处理。例如矩阵组装，数值积分，线性代数处理等等。

如果采用 Python 接口，只需提供一个 .py 文件；如果采用 C++ 接口，需提供 .ufl 和 .cc 文件。大多数 FEniCS 提供的 C++ 库都可以通过 python 来调用，故而使用 python 和 C++ 在效率上可能不会差太多。本项目中选择使用 python 编写有限元计算程序。

4.1.2 FEniCS 程序示例

一个完整的 FEniCS 程序主要包括调用所需的库、定义网格、定义函数空间、定义边界以及边界条件，建立控制方程，求解控制方程，输出结果以及后处理部分，接着以一个超弹性长方体单轴拉伸的程序为例，简单介绍一下 FEniCS 程序的编写运行以及后处理部分。

超弹性长方体的尺寸为长 100cm，宽 5cm，高 5cm，长方体左侧固定 x 方向位移，后面固定 y 方向位移，底面固定 z 方向位移，沿着方向单轴拉伸 150cm，超弹性体选用不可压缩 Neo-Hookean 模型。

解决此类边界值问题，FEniCS 包括以下步骤：

1. 调用必要的 python 库函数；
2. 识别计算域 (Ω)，偏微分方程，它的边界和边界条件。
3. 将偏微分方程重新定义为有限元变分问题。
4. 编写一个定义计算域的 Python 程序，变分问题，初始条件和边界条件等，使用相应的 FEniCS 对象进行抽象。
5. 调用 FEniCS 来解决边界值问题，并且可以扩展程序计算衍生量，如通量和平均值等，以及使用 Paraview 等库进行后处理以及展示可视化结果。

Python 程序首先调用需要使用的库函数，如下所示 (文章所有 Python 程序中提及的变量与正文中公式对应变量对照表格参考附录 A)：

```
1. import matplotlib.pyplot as plt
2. from fenics import *
3. from mshr import *
4. import numpy as np
```

接下来生成简单的几何体以及对应的网格，将长方体按照 $100 \times 5 \times 5$ 划分网格，如程序所示，可以使用 mshr 库来定义网格，也可以直接定义一些简单几何的网格：

```
7. T = 1.0
8. num_steps = 5
9. dt = T/num_steps
10. # Optimization options for the form compiler
11. parameters["form_compiler"]["cpp_optimize"] = True
12. parameters["form_compiler"]["representation"] = "uflacs"
13. # Create mesh and define function space
```

```

14. nx = nz = 5
15. ny = 100
16. #domain = Box(Point(0.0,0.0,0.0),Point(5.0,100.0,5.0))
17. #mesh = generate_mesh(domain, 16)
18. mesh = BoxMesh(Point(0.0,0.0,0.0),Point(5.0,100.0,5.0),nx,ny,nz)

```

一旦创建了网格，接着需要定义有限元的计算空间（域）VectorElement, FiniteElement 等，其中 VectorElement 代表向量空间，FiniteElement 代表标量空间。最终的计算域 FunctionSpace。FEniCS 支持的单元和符号定义可参考 <http://femtable.org/femtable.pdf>:

```

19. V = VectorElement('P', tetrahedron, 2)
20. Q = FiniteElement('P', tetrahedron, 1)
21. TH = V*Q
22. W = FunctionSpace(mesh,TH)

```

下一步是定义边界和边界条件，FEniCS 中用于定义 Dirichlet 边界条件命名为 DirichletBC，W 计算域的 sub(0)为位移空间，sub(0).sub(0)为 x 方向的位移，以此类推。位移边界条件可以用 Expression 对象来定义，将所有的边界条件赋值给变量空间 bcs:

```

23. # Mark boundary subdomains
24. left = CompiledSubDomain("near(x[1], side) && on_boundary", side = 0.0)
25. right = CompiledSubDomain("near(x[1], side) && on_boundary", side = 100.0)
26. top = CompiledSubDomain("near(x[2], side) && on_boundary", side = 5.0)
27. bottom = CompiledSubDomain("near(x[2], side) && on_boundary", side = 0.0)
28. front = CompiledSubDomain("near(x[0], side) && on_boundary", side = 5.0)
29. behind = CompiledSubDomain("near(x[0], side) && on_boundary", side = 0.0)
30. # Define boundary conditions
31. u1 = Expression("150*t", degree = 1, t = 0)
32. bcl = DirichletBC(W.sub(0).sub(1), Constant(0), left)
33. bcr = DirichletBC(W.sub(0).sub(1), u1, right)
34. bcb = DirichletBC(W.sub(0).sub(2), Constant(0), bottom)
35. bcd = DirichletBC(W.sub(0).sub(0), Constant(0), behind)
36. bcs = [bcl, bcr, bcb, bcd]

```


接下来定义各个变量及变量的试函数， u ， v 分别是位移及其对应的试函数， p, q 分别是拉氏乘子及其对应的试函数， F 和 F_v 分别是变形梯度及其对应的试函数， B 和 T 分别是体力和面力：

```

37. # Define test functions
38. w = Function(W)
39. u, p = split(w)
40. (v, q) = TestFunctions(W)
41. wt = TestFunction(W)
42. dw = TrialFunction(W)
43. # Define deformation gradient tensor
44. F = grad(u) + Identity(len(u))
45. Fv = grad(v)
46. B = Constant((0.0, 0.0, 0.0)) # Body force per unit volume
47. T = Constant((0.0, 0.0, 0.0)) # Traction force on the boundary
    
```

接下来计算定义一些相关的量，如右柯西格林张量，及其三个主不变量；定义材料的杨氏模量和泊松比，并计算对应的拉梅常数：

```

48. C = F.T*F # Right Cauchy-Green tensor
49. C_2 = C*C
50. # Invariants of deformation tensors
51. Ic = tr(C)
52. Ic2 = 0.5*(tr(C)*tr(C)-tr(C_2))
53. J = det(F)
54. # Elasticity parameters
55. E, nu = 10.0, 0.3
56. mu, lmbda = Constant(E/(2*(1 + nu))), Constant(E*nu/((1 + nu)*(1 - 2*nu))
    
```

接着推导超弹性体变形的控制方程，利用变分法进行如下推导：

不可压缩 neo-Hookean model 总的势函数如下：

$$\begin{aligned} \Pi[\mathbf{x}, p] = & \int \left[\frac{\mu}{2} (F_{iK} F_{iK} - 3) - p (\det \mathbf{F} - 1) \right] dV \\ & - \int B_i x_i dV - \int T_i x_i dA \end{aligned} \quad (4-1)$$

在平衡条件下，由最小势能原理可得：

$$\begin{aligned}
 \delta \Pi &= \int [\mu(F_{iK} \delta F_{iK}) - \delta p(\det \mathbf{F} - 1) - p \delta(\det \mathbf{F})] dV - \int B_i \delta x_i dV - \int T_i \delta x_i dA \\
 &= \int [\mu(F_{iK} \delta F_{iK}) - \delta p(\det \mathbf{F} - 1) - p \det \mathbf{F} H_{Ki} \delta F_{iK}] dV - \int B_i \delta x_i dV - \int T_i \delta x_i dA \\
 &= \int \left[\mu \left(F_{iK} \frac{\partial \delta x_i}{\partial X_K} \right) - \delta p(\det \mathbf{F} - 1) - p \det \mathbf{F} H_{Ki} \frac{\partial \delta x_i}{\partial X_K} \right] dV - \int B_i \delta x_i dV - \int T_i \delta x_i dA \quad (4-2) \\
 &= 0
 \end{aligned}$$

其中 \mathbf{H} 是变形梯度 \mathbf{F} 的逆张量：

$$\frac{\partial \det \mathbf{F}}{\partial F_{iK}} = \det \mathbf{F} H_{Ki} \quad (4-3)$$

其他超弹性体模型也可以通过上述类似的推导过程得到控制方程。

弱形式控制方程还可以通过 FEinCS 中的 `derivative` 函数直接作用于势函数求得，`derivative` 函数作用于弱形势 `a` 可以实现雅可比矩阵的实时更新：

```

57. # Define the weak form (incompressible neo-Hookean)
58. a = mu*inner(F, Fv)*dx - p*det(F)*inner(inv(F).T, Fv)*dx
59.     - (det(F)-1)*q*dx
60. # Stored strain energy density (incompressible neo-Hookean model)
61. psi = (mu/2)*(tr(F.T*F) - 3) - p*(det(F)-1)
62. # Total potential energy
63. Pi0 = psi*dx - dot(B, u)*dx - dot(T, u)*ds
64. a = derivative(Pi0, w, wt)
65. J1 = derivative(a, w, dw)
    
```

接下来更新每一步的时间及对应的位移场，代入边界条件 `bcs` 和雅可比矩阵求解控制方程，并输出结果 `vtkfile`。

```

66. t = 0.0
67. for n in range(num_steps):
68.     # Update current time
69.     t += dt
70.     u1.t = t
71.     print(t)
72.     solve(a == 0, w, bcs, J=J1)
73.     vtkfile << (w.u, t)
    
```

简单的 `plot` 命令对于快速可视化是有用的，但是对于更高级的可视化必须

使用外部工具。例如可以使用 ParaView 可视化解决方案，ParaView 的详细操作可以参考 <http://www.paraview.org>，结果如图 4-1 所示：

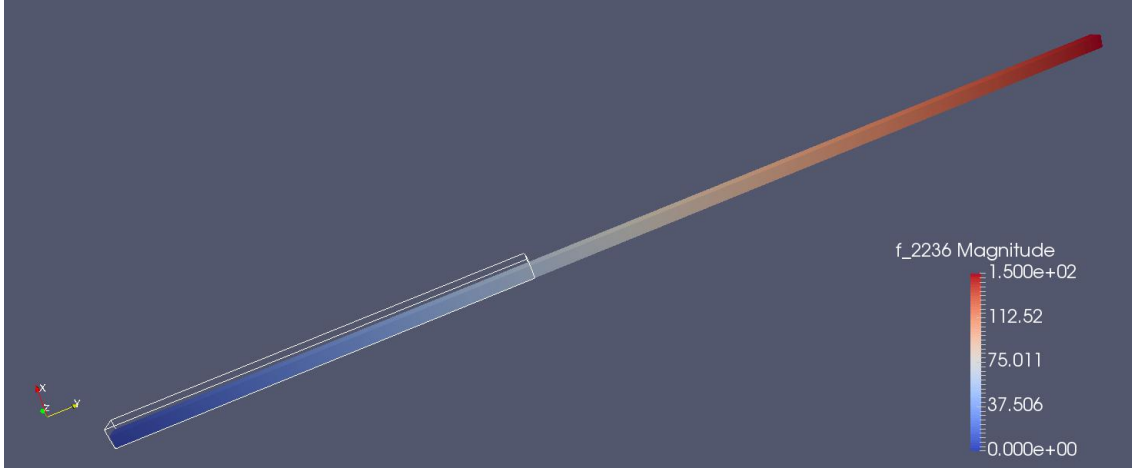


图 4-1 超弹性体单轴拉伸位移场结果 ParaView 可视化

4.2 水凝胶的溶胀 FEniCS 程序

4.2.1 搭建 FEniCS 水凝胶溶胀模型

参考章节 3.1 中的第四方式来搭建水凝胶溶胀的 FEniCS 程序，将控制方程转化为 FEniCS 程序语言：

程序中可能会使用的一些常量：

$$\Omega = v = 10^{-28} \text{ m}^3, kT = 4 \times 10^{21} \text{ J}$$

$$\frac{kT}{\Omega} = 4 \times 10^7 \text{ Pa}, NkT = 10^5 \text{ N/m}^2$$

$$Nv = 0.001, \chi = 0.2$$

考虑粘性耗散项，根据公式(3-54)：

$$\begin{aligned} \dot{W}_{vis} &= \eta \det \mathbf{F} \left[\frac{1}{2} \frac{\partial v_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{1}{3} \left(\frac{\partial v_k}{\partial x_k} \right)^2 \right] \\ &= \eta \det \mathbf{F} \left[\frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} \right)^2 + \frac{1}{2} \frac{\partial v_i}{\partial x_j} \frac{\partial v_j}{\partial x_i} - \frac{1}{3} \left(\frac{\partial v_k}{\partial x_k} \right)^2 \right] \end{aligned}$$

Python 程序首先调用需要使用的库函数时间相关问题设置合适的时间步，并设置编译器优化选项；接着生成简单的几何体以及对应的网格，将长宽都为 20 的矩形划分为 20×20 的网格，然后需要设置对应的边界条件：

```

1. nx = 40
2. ny = 40
3. mesh = RectangleMesh(Point(0.0,0.0),Point(20.0,20.0), nx, ny)
4. # Define boundary conditions
5. mu_D = Expression("mu00*(1-t/100)", degree = 1, mu00 = mu10, t = 0)
6. #fixed left and bottom sides
7. bcl = DirichletBC(W.sub(0).sub(0), Constant(0),left)
8. bcb= DirichletBC(W.sub(0).sub(1), Constant(0),bottom)
9. bct_mu = DirichletBC(W.sub(1), mu_D,top)
10. bcr_mu = DirichletBC(W.sub(1), mu_D, right)
11. bcs = [bcl,bcb,bct_mu,bcr_mu]
    
```

接下来定义各个变量及变量的试函数， u, v 分别是位移及其对应的试函数， μ 、 μ_0 分别是化学势及其对应的试函数， F 和 F_v 分别是变形梯度及其对应的试函数， H 是 F 的逆张量，程序核心部分是控制方程偏微分方程的定义以及求解：

```

12. #initial chemical potential value in the domain
13. e_u0 = Expression(('0', '0'), degree=1)
14. e_mu0 = Expression('mu0', mu0=mu0, degree=1)
15. assign(w, [interpolate(e_u0, V1), interpolate(e_mu0, Q1)])
16. ...
17. dt = Expression('dt1', degree=0, dt1=0.1)
18. gradv = grad((u-u_n)/dt)
19. W_vis = vis*det(F)*(0.5*inner(gradv*H, (gradv*H+H.T*gradV.T)) -\
20.      1/3* tr(gradv*H)* tr(gradv*H))
21. W_FC = 0.5*Nv*(lambda0**2*Ic-3-2*ln(lambda0**3*det(F)))-\
22.      C1*(ln(1+1/C1))-chi/(1+C1)
23. Pi0 = -W_FC - W_vis*dt
24. a0 = (derivative(Pi0, w, wt) - W_vis*dt + mu*derivative(C1, w, wt))*dx
25.      (-1*D*C1*dot(H.T*grad(mu),H.T*grad(mu_))
26.      - lambda0**3*det(F)*inner(gradv,H.T)*mu_)*dx
27. J1 = derivative(a0, w, dw)
    
```

二维水凝胶溶胀的位移云图及化学势场结果如下图所示：

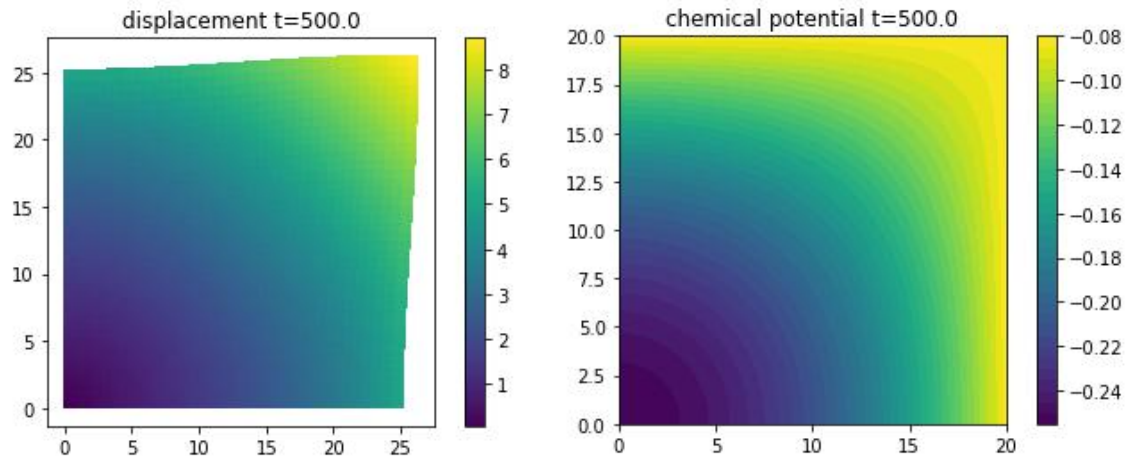


图 4-2 二维水凝胶溶胀模型位移场和化学势场结果

完整的 FEniCS 水凝胶 python 程序见附录 B。

4.2.2 网格尺寸的收敛性分析

为了得到合理的数值解，需要对上述水凝胶模型使用不同尺寸的单元进行划分，进行网格尺寸的收敛性分析。主要选取了网格尺寸为 4、2、1、0.5 以及 0.25 的模型，从运行结果中选取 $t=1$ 、 $t=10$ 、 $t=50$ 、 $t=100$ 以及 $t=500$ 时刻的最大位移的误差作为参考量进行比较，如下表 4-1 所示：

表 4-1 网格尺寸的收敛性分析（最大位移的误差作为比较量）

网格尺寸	$t=1$	$t=10$	$t=50$	$t=100$	$t=500$
4	0	0	0	0	0
2	1.66E-03	6.86E-03	0.026	0.167	0.005
1	2.49E-03	8.79E-03	0.033	0.207	0.007
0.5	2.50E-03	9.34E-03	0.035	0.22	0.007
0.25	2.50E-03	9.47E-03	0.0354	0.223	0.007

将上述表格中的数据绘制成如下收敛曲线图，可以更直观地进行网格尺寸的收敛性分析：

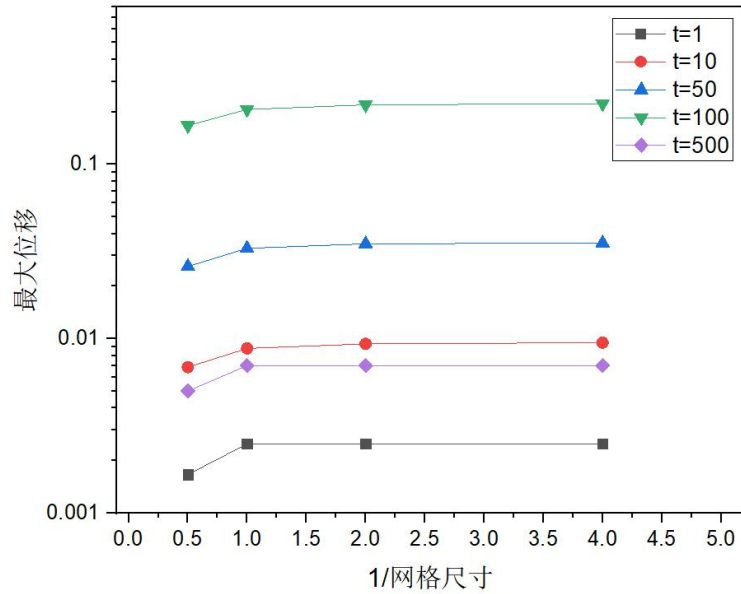


图 4-3 网格尺寸的收敛性分析（最大位移的误差作为比较量）

根据表 4-1 以及图 4-3 中的结果可知，当网格尺寸小于 0.5 时，有限元结果趋向于收敛，综合数值解的合理性以及计算成本的考量，二维水凝胶溶胀模型选择网格尺寸为 0.5。

4.2.3 时间步长的收敛性分析

根据上述 4.2.2 中的结果，选择网格尺寸为 0.5 的模型进行时间步长的收敛性分析。主要选取了时间步长 dt 为 4、2、1、0.5 以及 0.25（时间段 1~90 区间的时间步长，0~1 区间时间步长统一选择 0.1，90~500 区间时间步长统一选择 5）的模型，从运行结果中选取 $t=1$ 、 $t=10$ 、 $t=50$ 、 $t=100$ 以及 $t=500$ 时刻的最大位移的误差作为参考量进行比较，如下表 4-2 所示：

表 4-2 时间步长的收敛性分析（最大位移的误差作为比较量）

时间步长 dt	$t=1$	$t=10$	$t=50$	$t=100$	$t=500$
4	0	0	0	0	0
2	0	3.10E-03	0.01	0.022	0.003
1	0	4.00E-03	0.013	0.02	0.005
0.5	0	5.50E-03	0.017	0.038	0.006
0.25	0	6.00E-03	0.019	0.041	0.006

将上述表格中的数据绘制成如下收敛曲线图，可以更直观地进行时间步长的收敛性分析：

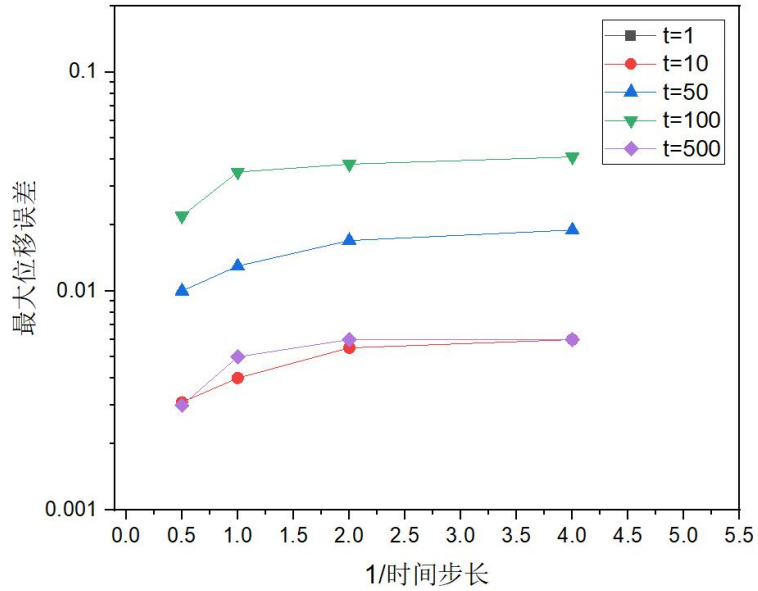


图 4-4 时间步长的收敛性分析（最大位移的误差作为比较量）

根据表 4-2 以及图 4-4 中的结果可知，当时间步长小于 0.5 时，有限元结果趋向于收敛，综合数值解的合理性以及计算成本的考量，二维水凝胶溶胀模型选择时间步长为 0.5。

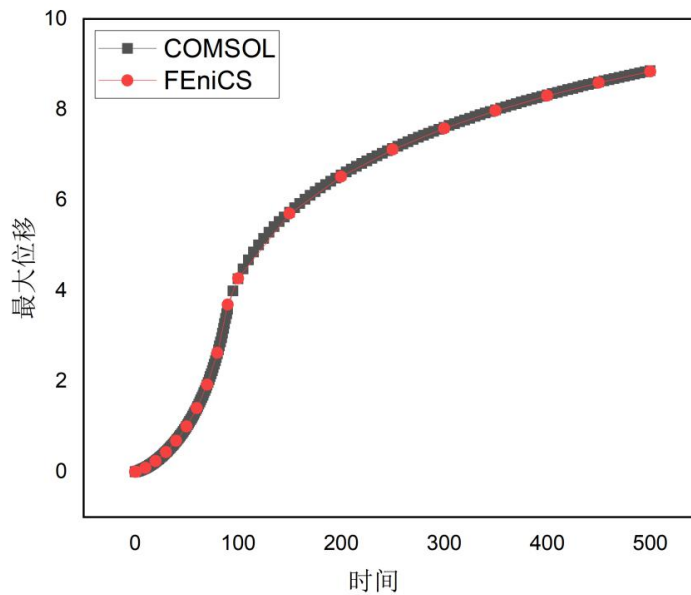


图 4-5 COMSOL 二维水凝胶溶胀模型与 FEniCS 模型最大位移比较

将上述网格尺寸为 0.5，时间步长为 0.5 的 FEniCS 模型最大位移结果与 COMSOL 二维水凝胶溶胀模型的结果作比较，如图 4-5 所示：

根据图 4-5 的结果，COMSOL 二维水凝胶溶胀模型与 FEniCS 模型最大位移结果基本一致，一定程度上说明利用 FEniCS 平台实现的二维水凝胶程序的正确性。以上述 FEniCS 二维水凝胶溶胀模型为基础，可以扩展研究其他水凝胶溶胀力学行为问题。

4.2.4 FEniCS 水凝胶溶胀混合单元模型

上述水凝胶溶胀的 FEniCS 程序，因为高分子网络与溶剂小分子的不可压缩性约束是通过公式(3-17)消去浓度场 C 显式得到，控制方程只需要求解位移场和化学势场。同样的约束也可以通过拉氏乘子法加以强制，此时系统需要求解位移场 \mathbf{u} 、拉氏乘子场 Π 、溶剂浓度场 C 以及化学势场 μ 四个未知函数。相应地，系统的控制方程变更如下：

(1) 溶剂分子守恒（根据公式(3-21)）：

$$\frac{1}{J_0} \int \left[-DCH_{K_i} \frac{\partial \mu}{\partial X_K} \frac{\partial \delta \mu}{\partial X_L} H_{L_i} - \frac{\partial C}{\partial t} \delta \mu \right] dV = 0 \quad (4-4)$$

(2) 力的平衡方程：（根据公式(3-24) W_s 和 W_{vis} 保持不变）：

$$W_m = C \ln \left(\frac{C}{\lambda_0^3 \det \mathbf{F}} \right) + \chi \frac{C}{\lambda_0^3 \det \mathbf{F}} \quad (4-5)$$

$$\frac{1}{J_0} \left(-\delta (W_s + W_m) + \mu \delta C - W_{vis} \right) + \delta \left(\Pi \left(\det \mathbf{F} - \frac{1+C}{J_0} \right) \right) = 0 \quad (4-6)$$

在对应的 FEniCS 程序中，需要在章节 4.2.1 模型的基础上，对解空间以及弱形式相应修改如下：

```

1. V = VectorElement('P',triangle,2)
2. Q = FiniteElement('P',triangle,1)
3. TH = MixedElement([V, Q, Q, Q])
4. W = FunctionSpace(mesh,TH)
5. # Define test functions
6. w = Function(W)
7. u, Pi, Cs, mu = split(w)
8. ...
9. #initial chemical potential value in the domian
10. e_u0 = Expression(('0', '0'), degree=1)
```



```

11. e_Pi0 = Expression('0', degree=1)
12. e_C0 = Expression('C0', C0 = C0, degree = 1)
13. e_mu0 = Expression('mu0', mu0=mu0, degree=1)
14. assign(w, [interpolate(e_u0, V1), interpolate(e_Pi0, Q1), \
15.             interpolate(e_C0, Q1), interpolate(e_mu0, Q1)])
16. W_FC= 0.5*Nv*(lambda0**2*Ic-3-2*ln(lambda0**3*det(F)))\
17.       - Cs*(ln(1+1/Cs))-chi/(1+Cs) + Pi*(det(F)-(1+Cs)/J0)

```

将上述网格尺寸为 0.5，时间步长选择 1 的多场耦合模型最大位移结果与章节 4.2.1 中的 FEniCS 模型以及 COMSOL 二维水凝胶溶胀模型的结果作比较，如图 4-6 所示：

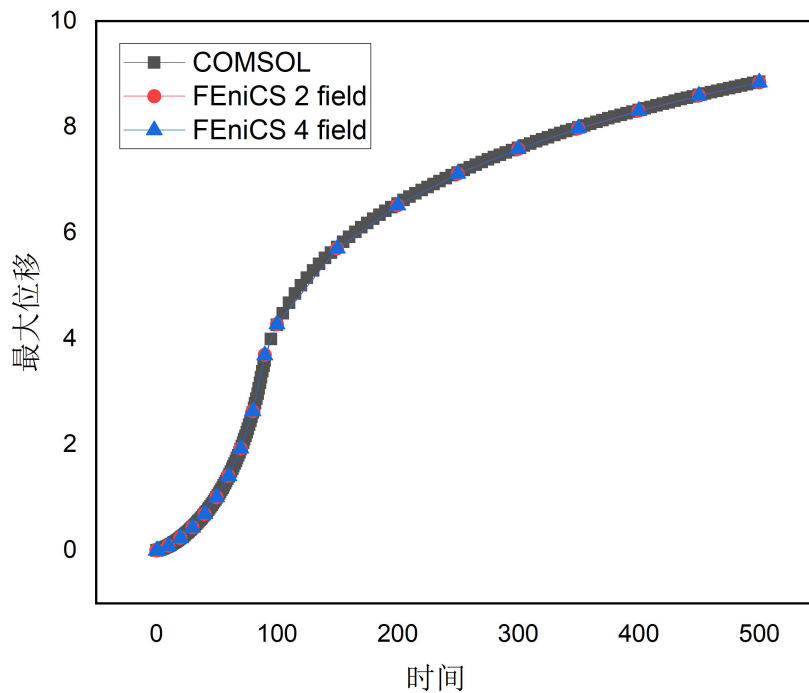


图 4-6 COMSOL 二维水凝胶溶胀模型与 FEniCS 两种模型最大位移比较

根据图 4-6 中的结果可知，混合单元模型最大位移结果与 COMSOL 模型，以及 FEniCS 简单的水凝胶双场耦合模型的结果基本一致。

4.3 FEniCS 化学凝胶溶胀力学行为研究

4.3.1 基于标度率修正的化学凝胶溶胀模型

水凝胶在脱水的过程中弹性模量会增加，当凝胶吸水时，弹性模量会变低。根据章节 3.3 中的标度率，水凝胶在溶胀和退溶胀过程中弹性模量与高分子的体积分数之间的关系为：

$$E = E_{ini}\phi^{0.56}, \text{ 溶胀}$$

$$E = E_{ini}\phi^{2.3}, \text{ 退溶胀}$$

在溶胀过程中，水凝胶的弹性能密度根据公式(3-66)：

$$W_s(F) = \frac{1}{2} \frac{NkT}{\lambda_0^3} \left(\left(\frac{\lambda_0}{\lambda_{ref}} \right)^2 F_{iK} F_{iK} - 3 - 2 \ln(\det F) - 6 \ln \left(\frac{\lambda_0}{\lambda_{ref}} \right) \right)$$

此时，在 FEniCS 程序中只需要给出高分子的体积分数 ϕ ，以及高分子网络由溶胀引起的松弛参考构型拉伸比 λ_{ref} ，然后修改弹性能密度方程即可。

1. C0 = 0.2
2. v_const = 0.588
3. vttotal = -(3*v_const**2 - 2.5*v_const + 0.5)
4. phi0v = (1/(C0+1))**vttotal
5. C1 = lambda0**3*det(F) - 1
6. phi = 1/(1+C1)
7. lambda_ref = (phi**vttotal)/phi0v
8. W_FC = 0.5*Nv*(lambda0**2*Ic/(lambda_ref**2)-3 \
9. -2*ln(lambda0**3*det(F)/(lambda_ref**3))) \
10. - C1*(ln(1+1/C1))-chi/(1+C1)

在退溶胀过程中，水凝胶的弹性能密度根据公式(3-70)：

$$W_s(F) = \frac{1}{2} \frac{NkT}{\lambda_0^3} \left(\left(\frac{1}{\lambda_{ref}} \right)^2 F_{iK} F_{iK} - 3 - 2 \ln(\det F) + 6 \ln \lambda_{ref} \right)$$

其中 N 与溶胀率的关系如式(3-74)所给定。此时在 FEniCS 程序中只需要对应地修改化学势边界条件和弹性能密度表达式即可：

1. mu_D = Expression("mu00*(1 + t/50)", degree = 1, mu00 = mu10, t = 0)

```

2. ...
3. C0 = 0.2
4. v_const = 0.588
5. vtotal = -(3*v_const**2 - 2.5*v_const + 0.5)
6. phi0v = (1/(C0+1))*vtotal
7. C1 = lambda0**3*det(F) - 1
8. phi = 1/(1+C1)
9. lambda_ref = (phi**vtotal)/phi0v
10. W_FC = 0.5*Nv*(phi*1.2)**1.07*(Ic/(lambda_ref**2)-3 \
11.      -2*ln(det(F)/(lambda_ref**3))) \
12.      - C1*(ln(1+1/C1))-chi/(1+C1)
    
```

FEniCS 溶胀退溶胀模型运行的结果，位移场如图 4-7 所示：

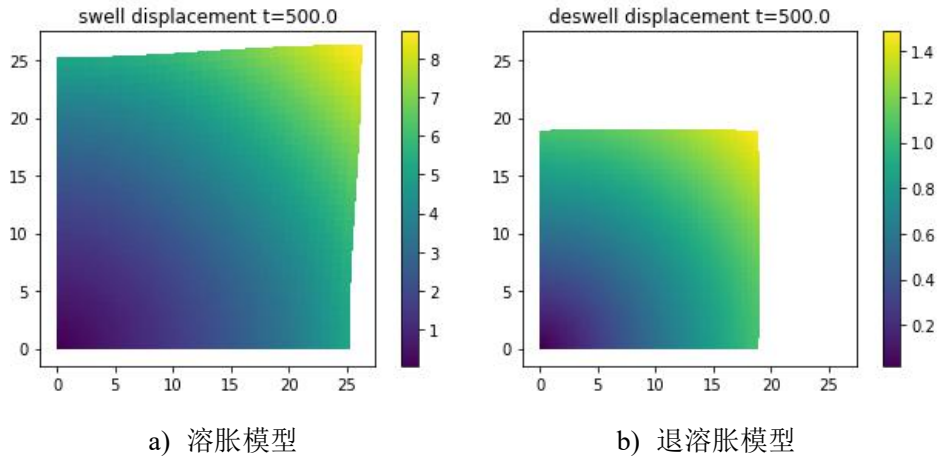


图 4-7 FEniCS 水凝胶 Scaling Theory 模型运行的位移场结果(a)溶胀模型；(b)退溶胀模型

4.3.2 水凝胶薄膜的起皱现象

在日常生活中，很多人都会注意到，洗澡的时候长期浸泡在水中的手掌会起皱，此外皮肤老化、未完全固化的涂料和老化的壁纸都是生活中常见的起皱(褶皱)现象。起皱是由具有不同机械性能的材料组成的表面发生不均匀膨胀导致，当材料中的内应力超过临界值时，褶皱就会出现。加热、吸收溶胀、机械拉伸都能导致褶皱的产生。

如图 4-8 所示，一个水凝胶双层结构，上层薄膜与底层基底水凝胶的模量不同，薄膜的弹性模量远高于基底。在加热或者吸水溶胀的情况下，当薄膜中的内应力大于临界弯曲应力之后，褶皱就会出现。

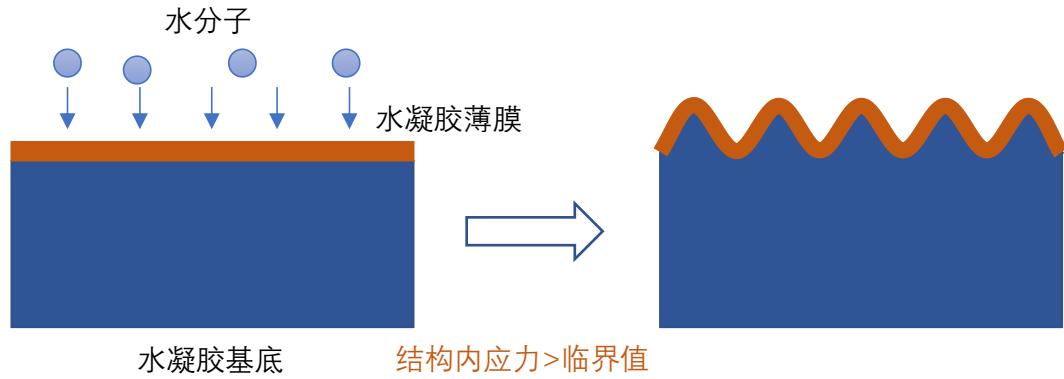


图 4-8 水凝胶双层结构褶皱出现的原理图^[113]

接下来设计一个溶胀导致水凝胶双层结构起皱的模型，并使用 FEniCS 程序来实现。设计如图 4-9 所示的水凝胶双层结构，基底的尺寸为 100×50 ，薄膜的厚度与基底厚度比为 2:50，基底水凝胶的无量纲弹性模量为 0.001，薄膜的弹性模量基底的弹性模量比 1000:1。在初始状态下，基底和薄膜中的水分子浓度都为 $0.2/(1+0.2)$ 。在该溶胀模型中，几何的边界均为零通量边界条件，底边固定住 y 方向的位移，左右两边固定住 x 方向的位移。

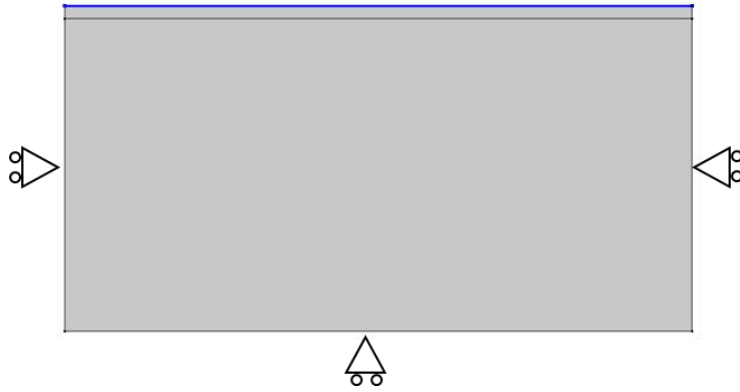


图 4-9 水凝胶双层结构起皱的几何模型

在 FEniCS 中实现上述模型，需要建立一个 200×52 的矩形，然后划分上下两块区域，对不同的区域分别赋予不同的弹性模量 Nv （薄膜为 1，基底为 0.001）、相同的水分子相对体积（薄膜为 0.1，基底为 0.1）、不同的化学势（由水分子相对体积计算得出薄膜为 -0.712，基底为 -0.819），然后规定对应的边界条件即可，程序的其他部分与章节 4.1 中二维水凝胶的溶胀模型基本一致。

1. #Initialize C0 Nv mu0
2. tol = 1E-14

```

3. C0_0 = 0.1
4. C0_1 = 0.1
5. Nv_0 = 0.001
6. Nv_1 = 1
7. vis = 0.01
8. D = 5          #diffusion constant
9. #define different C0 Nv mu0 in defferent domains
10. C0 = Expression('x[1]<=50.0 + tol ? C0_0 : C0_1', degree=0, tol = tol, \
11.          C0_0 = C0_0, C0_1 = C0_1)
12. Nv = Expression('x[1]<=50.0 + tol ? Nv_0 : Nv_1', degree=0, tol = tol, \
13.          Nv_0 = Nv_0, Nv_1 = Nv_1)
14. mu0 = Expression('x[1]<=50.0 + tol ? mu_0 : mu_1', degree=0, tol = tol, \
15.          mu_0 = mu_0, mu_1 = mu_1)
16. lambda0 = (1+C0)**(1/3)
17. ....
18. #fixed left and bottom sides
19. bcl = DirichletBC(W.sub(0).sub(0), Constant(0), left)
20. bcr = DirichletBC(W.sub(0).sub(0), Constant(0), right)
21. bcb = DirichletBC(W.sub(0).sub(1), Constant(0), bottom)
22. bcs = [bcl,bcr,bcb]

```

FEniCS 程序的结果如图 4-10 所示:

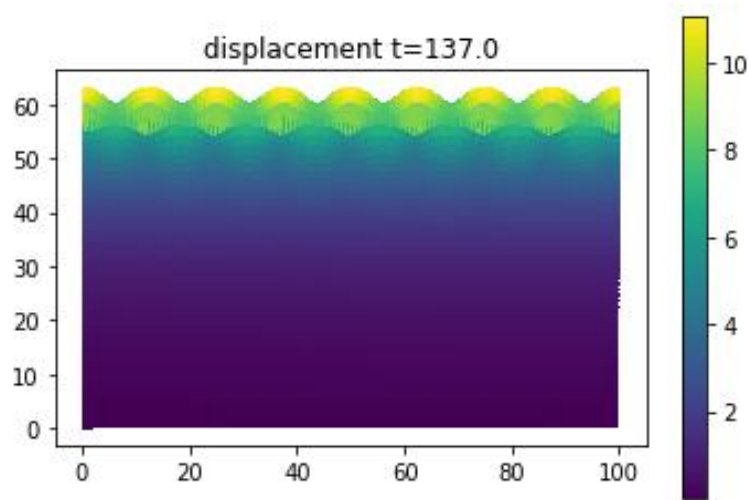


图 4-10 水凝胶双层结构起皱模型的位移场

在上述程序的结果中，由于水分子的扩散，水凝胶双层结构中的上层薄膜吸水溶胀，表层薄膜最先溶胀体积增大，薄膜横向受压，当应力超过某一临界值时，褶皱产生。随着时间的变化，水凝胶双层结构中的水分子浓度及化学势会逐渐均匀分布，最终形成稳定尺寸和形状的褶皱。

4.3.3 3D 打印凝胶长条溶胀行为研究

本课题组在使用 3D 打印技术制备水凝胶的过程中发现，打印出的水凝胶长条浸入水中吸水溶胀时，不对称的月牙形横截面会导致水凝胶发生弯曲，水和原始溶剂的扩展速率不同又会导致弯曲方向的交替。弯曲曲率和响应速率可以通过改变 3D 打印参数来调整。

如图 4-10 所示为上述截面不对称水凝胶长条浸入水中吸水溶胀发生交替弯曲的试验以及仿真结果图。上述实验是一个非常有趣的化学凝胶溶胀动力学行为。下面的内容将会用理论来解释上述溶胀行为，并使用 FEniCS 进行相关的建模。

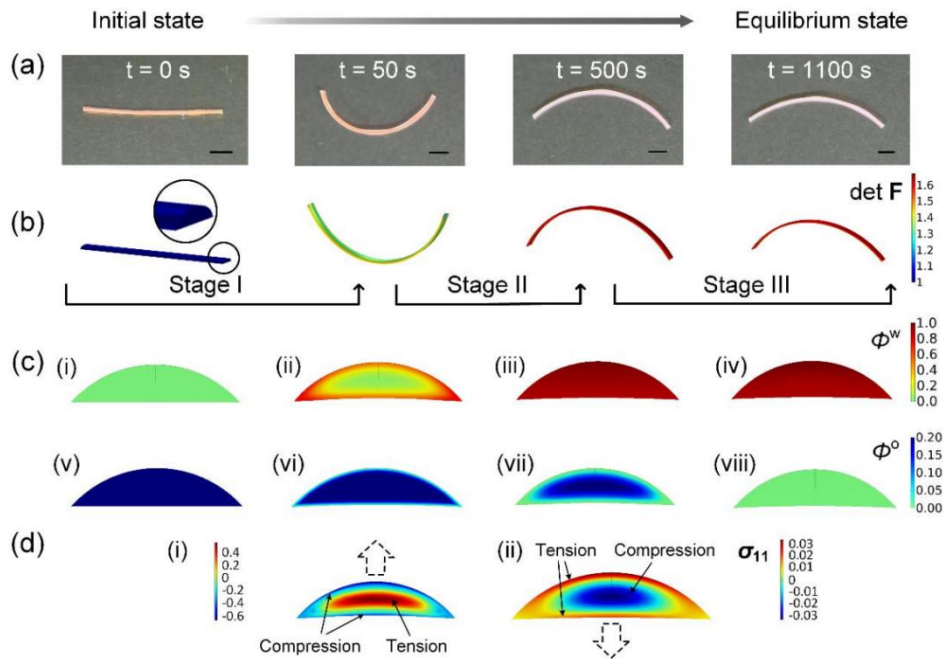


图 4-11 截面不对称水凝胶长条的动力学弯曲试验及 COMSOL 仿真^[67]

有机溶剂的扩展速率要远小于水的扩散速率，因此当水凝胶长条状浸入水中，环境中的水分子会快速扩散到凝胶中，如图 4-10C(ii)所示，此时凝胶表层的溶胀程度比较高，体积发生膨胀。这将导致水凝胶的表层受压，而内部受拉。而截面的中心和其形心通常不重合，在内外受拉压状态不一致的情况下导致弯矩的产生。

月牙形的截面，其形心靠近较平的一侧，因此对应的较弯一侧受到的压力会产生更大的弯矩，使得水凝胶向较弯的一侧发生弯曲变形。但是当水分子扩散到长条的内部，使的水凝胶内外的溶胀程度一致时，由水分子扩散导致的弯矩将会逐渐消失，长条会慢慢由弯变直。但此时有机溶剂向环境中扩展产生的效应又慢慢显现出来，有机溶剂的扩散导致凝胶的外表面发生收缩，导致外部受拉内部受压，此时水凝胶的截面形状并没有发生很大的改变，其截面的形心还是靠近较平的一侧，因此对应的较弯一侧受到的拉力又会产生更大的弯矩，使得水凝胶又开始向较平的一侧发生弯曲变形。

上述的凝胶的溶胀和大变形耦合理论同样可以总结为溶剂分子守恒以及力的平衡方程两个控制方程：

(1) 溶剂分子守恒(根据公式(3-21)):

$$\begin{cases} \frac{1}{J_0} \int \left[-D_1 C_1 H_{Ki} \frac{\partial \mu_1}{\partial X_K} \frac{\partial \delta \mu_1}{\partial X_L} H_{Li} - \frac{\partial C_1}{\partial t} \delta \mu_1 \right] dV = 0 \\ \frac{1}{J_0} \int \left[-D_2 C_2 H_{Ki} \frac{\partial \mu_2}{\partial X_K} \frac{\partial \delta \mu_2}{\partial X_L} H_{Li} - \frac{\partial C_2}{\partial t} \delta \mu_2 \right] dV = 0 \end{cases} \quad (4-7)$$

(2) 力的平衡方程: (根据公式(3-24) W_s 和 W_{vis} 保持不变)

$$\begin{aligned} W_m = & C_1 \ln \left(\frac{C_1}{\lambda_0^3 \det \mathbf{F}} \right) + C_2 \ln \left(\frac{C_2}{\lambda_0^3 \det \mathbf{F}} \right) + \chi_{1p} \frac{C_1}{\lambda_0^3 \det \mathbf{F}} \\ & + \chi_{2p} \frac{C_2}{\lambda_0^3 \det \mathbf{F}} + \chi_{12} C_1 \frac{C_2}{\lambda_0^3 \det \mathbf{F}} \end{aligned} \quad (4-8)$$

$$\frac{1}{J_0} \left(-\delta (W_s + W_m) + \mu_1 \delta C_1 + \mu_2 \delta C_2 - W_{vis} \right) + \delta \left(\Pi \left(\det \mathbf{F} - \frac{1 + C_1 + C_2}{J_0} \right) \right) = 0 \quad (4-9)$$

上述公式中 C_1 、 C_2 、 μ_1 、 μ_2 分布为两种不同溶剂的浓度及化学势， χ_{1p} 、 χ_{2p} 、 χ_{12} 为不同溶剂与高分子网络，以及溶剂之间的无量纲混合焓， D_1 、 D_2 分别为两种溶剂的无量纲扩散系数。

参考章节 3.2 对上述溶胀模型进行速度的离散化，将上述推导的控制方程转化为 FEniCS 语言，程序中可能会使用的一些常量：

$$\Omega = v = 10^{-28} \text{ m}^3, kT = 4 \times 10^{21} \text{ J}, \frac{kT}{\Omega} = 4 \times 10^7 \text{ Pa}, \text{vis} = 0.01$$

$$NkT = 10^6 \text{ N/m}^2$$

$$C_{10} = 10, C_{20} = 0.001, C_{1final} = 0.001, C_{2final} = 10$$

$$D_1 = 1, D_2 = 0.1$$

$$J_0 = 1 + C_{10} + C_{20} = 11.001$$

$$N_v = 0.01, \chi_{1p} = \chi_{2p} = \chi_{12} = 0$$

Python 程序的主干部分，如下所示，时间相关问题设置合适的时间步，并设置编译器优化选项；外部导入划分好网格的几何(截面为月牙形的长条，长度为 48)。定义一些需要使用的常数参数，包括两种溶剂的初始浓度 C_{10} 、 C_{20} ，最终浓度 C_{1_final} 、 C_{2_final} ；初始化学势 μ_{10} 、 μ_{20} ，最终的化学势 μ_{1_final} 、 μ_{2_final} ；水凝胶初始时刻的体积 J_0 ，最终溶胀到的体积 J_final ；溶胀预拉伸比 λ_{0} 、最终溶胀到的拉伸比 λ_{final} ；动力粘度系数 vis ，扩散系数 D_1 、 D_2 ，程序的核心部分为：

```

1. cs1 = Cs1/(det(F)*lambda0**3)
2. cs2 = Cs2/(det(F)*lambda0**3)
3. C1t = (Cs1 - Cs1_n)/dt
4. C2t = (Cs2 - Cs2_n)/dt
5. gradv = grad((u-u_n)/dt)
6. W_vis = vis*det(F)*(0.5*inner(gradv*H, (gradv*H+H.T*gradV.T)) - \
7.      1/3* tr(gradv*H)* tr(gradv*H))
8. W_FC = 0.5*Nv*(lambda0**2*Ic-3-2*ln((lambda0)**3*det(F))) +
9.      Cs1*ln(cs1)+Cs2*ln(cs2)+chi1p*cs1+chi2p*cs2+chi12*Cs1*cs2
10. Pi0 = -W_FC -W_vis*dt
11. IMCOMPRESS = Pi*(det(F)-(1+Cs1+Cs2)/J0)
12. a0 = (derivative(Pi0, w, wt) + mu1*Cs1_ + mu2*Cs2_)/J0*dx
13. + derivative(IMCOMPRESS, w, wt)*dx+\
14. + (-1*D1*Cs1/J0*dot(H.T*grad(mu1),H.T*grad(mu1_)) - C1t/J0*mu1_)*dx \
15. + (-1*D2*Cs2/J0*dot(H.T*grad(mu2),H.T*grad(mu2_)) - C2t/J0*mu2_)*dx
16. J1 = derivative(a0, w, dw)
    
```

程序运行的结果如图 4-12 所示：

在上述程序的结果中，在溶剂分子扩散发生的短时间内水凝胶向较弯的一侧发生弯曲变形，随后又较快速地由弯变直；同时水凝胶长条的体积发生明显的膨胀；长条变直之后，又开始向较平的一侧发生弯曲变形，达到一定的弯曲程度后，又缓慢地由弯变直，最后水凝胶长条的体积也恢复到接近最初的状态。

上述具有不对称截面的 3D 打印凝胶长条随时间变化的溶胀行为是今后在使用 3D 打印制作水凝胶结构时需要注意的地方，同时也为设计具有刺激响应、电磁致动、形态随时间变化的智能水凝胶结构提供了一个很好的思路。

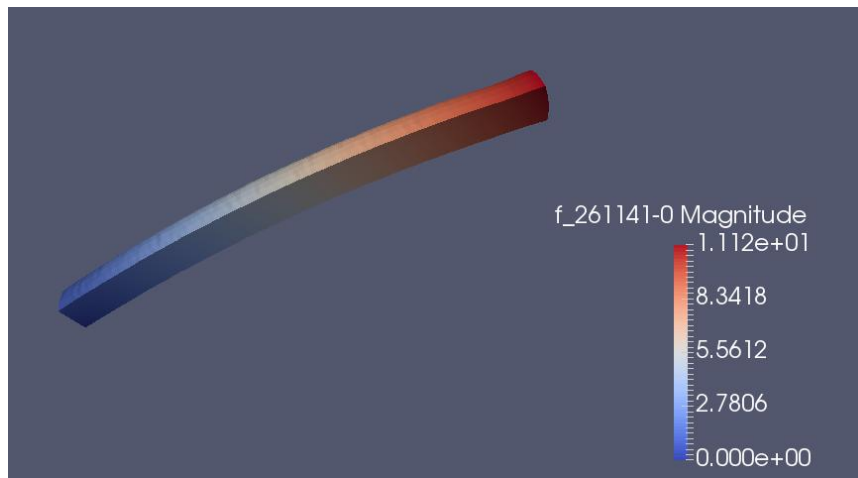


图 4-12 FEniCS 水凝胶长条溶胀动力学行为仿真结果

4.4 本章小结

第四章主要的内容包括：(1)简单介绍了 FEniCS 的功能、组件及各个组件的作用，以及结合示例对 FEniCS 程序组成进行了解释；(2)使用章节 3.1 中化学凝胶溶胀的控制方程，编写了化学凝胶溶胀二维模型的 FEniCS 程序，对部分程序进行了解释并与相应的 COMSOL 模型结果进行了比较；(3)将第三章中修正的标度律应用到化学凝胶溶胀 FEniCS 二维模型中，与理论以及 COMSOL 模型的结果进行比较；(4)结合 FEniCS 建模研究水凝胶薄膜溶胀过程中的起皱现象；(5)结合 FEniCS 建模研究不对称截面化学凝胶长条溶胀动力学行为。

结 论

本课题通过学习研究化学凝胶溶胀的非平衡态热力学理论模型、水凝胶在溶胀退溶胀过程中模量随高分子体积分数变化的标度律理论，并利用 COMSOL、FEniCS 建模研究了：(1)简单的 F-H 化学凝胶溶胀二维模型；(2)用标度律修正的化学凝胶溶胀、退溶胀二维模型中；(3)水凝胶双层结构溶胀起皱模型；(4)不对称截面化学凝胶长条溶胀动力学模型，得出了以下结论：

(1) 化学凝胶溶胀的过程是一个溶剂分子扩散动力学与凝胶宏观力学变形耦合的非平衡态热力学过程；

(2) 本课题从凝胶的自由能方程出发，给出了水凝胶在溶胀退溶胀条件下对应的自由能方程。结果显示，在溶胀过程中，水凝胶的弹性模量与聚合物体积分数 $\phi^{0.56}$ 成正比，在退溶胀过程中，水凝胶的弹性模量与 $\phi^{2.3}$ 成正比。上述理论与试验结果、COMSOL 中对应模型比较，结果均吻合；

(3) 在水凝胶双层结构溶胀起皱模型中，褶皱的特征尺度与薄膜与基底的弹性模量之比以及薄膜与基底的厚度之比相关；

(4) 在含有多种溶剂的水凝胶溶胀模型中，溶剂的扩散速率以及水凝胶几何上的不对称性会使凝胶发生动态的变形，这为制作水凝胶致动器以及搭建复杂的水凝胶结构提供了思路。

在过去的十年中，大量的研究人员致力于提高水凝胶性能，使其在生物医学、柔性电子、传感器和致动器等众多领域的应用机会大大增加。溶胀是水凝胶的一种常见现象，溶胀常常会导致水凝胶力学性能降低。为了解决水凝胶网络膨胀引起的机械不稳定性，研究水凝胶的溶胀的理论模型以及与力学性能之间的关系对于水凝胶的应用非常重要。将水凝胶制造与先进的生物制造技术相结合的应用前景也十分广阔。随着跨学科合作、多学科融合发展，将不同领域的先进技术结合起来，致力于提高水凝胶的力学性能、设计制造更实用的结构，引进一些新的功能，并结合其工程实际应用，推动水凝胶的进一步创新发展。

参考文献

- [1] 李录贤. 高等计算力学[M]. 西安交通大学出版社, 2018.
- [2] Permann, Cody, et al. MOOSE: Enabling Massively Parallel Multiphysics Simulation[J]. SoftwareX, 2020:100-430.
- [3] Gaston D R, Permann C J, Peterson J W, et al. Physics-based Multiscale Coupling for Full Core Nuclear Reactor Simulation[J]. Annals of nuclear energy, 2015, 84(oct.):45-54.
- [4] Bangerth W, Hartmann R, Kanschat G. Deal.II—A General-purpose Object-Oriented Finite Element Library[M]. ACM, 2007, 33(4):24-es.
- [5] Alzetta G, Arndt D, Bangerth W, et al. The Deal.II Library, Version 9.0[J]. Journal of numerical mathematics, 2018, 26(4):173-183.
- [6] Alnæs M, Blechta J, Hake J, et al. The FEniCS Project Version 1.5[J]. Archive of Numerical Software 3, 2015, 3(100).
- [7] Logg A, Mardal KA, Wells G. Automated Solution of Differential Equations by the Finite Element Method[M]. Springer Berlin Heidelberg, 2012, 24.
- [8] Hong W, Zhao X, Zhou J, Suo Z. A Theory of Coupled Diffusion and Large Deformation in Polymeric Gels[J]. Journal of the Mechanics & Physics of Solids, 56(5):1779-1793.
- [9] Li Z, Liu Z, Ng T Y, Sharma P. The Effect of Water Content on the Elastic Modulus and Fracture Energy of Hydrogel[J]. Extreme Mechanics Letters, 2019:100617.
- [10] Brandi M, Brocke E, Talukdar H A, et al. Connecting MOOSE and NeuroRD through MUSIC: towards a Communication Framework for Multi-Scale Modeling[J]. BMC Neuroscience, 2011, 12(Suppl 1):1-2.
- [11] Peterson J W, Lindsay A D, Kong F. Overview of the Incompressible Navier–Stokes Simulation Capabilities in the MOOSE Framework[J]. Advances in Engineering Software, 2018, 119:68-92.
- [12] Gaston D, Newman C, Hansen G, et al. MOOSE: A Parallel Computational Framework for Coupled Systems of Nonlinear Equations[J]. Nuclear Engineering & Design, 2009, 239(10):1768-1778.
- [13] Gamble K A, Williams A F, Chan P K, et al. A Feasibility Study on the Use of the MOOSE Computational Framework to Simulate Three-Dimensional Deformation of CANDU Reactor Fuel Elements[J]. Nuclear Engineering & Design, 2015, 293:385-394.
- [14] Bangerth W, Heister T, Heltai L, et al. The deal.II Library, Version 8.0[J]. 2013.
- [15] Bangerth W, Hartmann R, Kanschat G. Deal.II—A General-Purpose Object-Oriented Finite Element Library[J]. Acm Transactions on Mathematical Software,

- 2007, 33(4):24.
- [16] Joshi A, Thompson A, Sevickmuraca E M, et al. Adaptive Finite Element Methods for Forward Modeling in Fluorescence Enhanced Frequency Domain Optical Tomography[C]// Biomedical Topical Meeting. 2004.
- [17] Phunpeng V, Baiz P M. Mixed Finite Element Formulations for Strain-Gradient Elasticity Problems Using the FEniCS Environment[J]. Finite Elements in Analysis & Design, 2015, 96(C):23-40.
- [18] Hirshikesh, Natarajan S, Annabattula R K. A FEniCS Implementation of the Phase Field Method for Quasi-Static Brittle Fracture[J]. Frontiers of Structural & Civil Engineering, 2018(1-2):1-17.
- [19] Abert C, Exl L, Bruckner F, et al. magnum.fe: A Micromagnetic Finite-Element Simulation Code Based on FEniCS[J]. Journal of Magnetism & Magnetic Materials, 2013, 345(8):29-35.
- [20] Valensendstad K, Steinman D A. CFD Challenge: Solutions Using a Finite Element Method Flow Solver Implemented in FEniCS[J], 2012.
- [21] Lofink P, Müller W H. Numerical Modeling of Fluid-Structure Interaction of Blood in a Vein by Simulating Conservation of Mass and Linear Momentum with the Finite Element Method in FEniCS[J]. Pamm, 2013, 13(1):209-210.
- [22] 施桢华. 分布式面向对象多模型建模与仿真框架[J]. 情报指挥控制系统与仿真技术, 2003(6).
- [23] 赵博康. 基于面向对象方法的多物理场耦合框架与计算研究[D].2018.
- [24] 陈丽娟, 田兆斐, 赵博康, 邓硕. 竖直加热管道内干涸型临界流动沸腾数值分析[C]// 第十五届全国反应堆热工流体学术会议暨中核核反应堆热工水力技术重点实验室学术年会. 0.
- [25] 王晶晶. 弹性体中纳米缺陷的表面效应[D]. 2017.
- [26] 黄筑平. 连续介质力学基础[M]. 2012.
- [27] Bower A F. Applied Mechanics of Solids[M]. 2012.
- [28] Suo Z. Elasticity of Rubber-Like Materials[J]. 2013.
- [29] Thakur V K, Thakur M K. Hydrogels: Recent Advances[M]. 2018.
- [30] Zhang Y S, Khademhosseini A. Advances in Engineering Hydrogels[J]. Science, 2017, 356(6337):eaaf3627.
- [31] Campbell T D. Synthesis and Physical Characterization of Biocompatible Hydrogels[J]. Dissertations & Theses Gradworks, 2007.
- [32] Nemours E D. Methacrylate Resins[J]. Ind Eng Chem. 1936, 28:1160–1163.
- [33] Flory P J. Thermodynamics of High Polymer Solutions[J]. The Journal of Chemical Physics. 1941.
- [34] Huggins M L. Theory of Solutions of High Polymers1[J]. journal of the american chemical society, 1948, 64(2848):128.

- [35] Huggins, Maurice L. Properties of Rubber Solutions and Gels[J]. Industrial & Engineering Chemistry, 1943, 35(2):216-220.
- [36] Treloar L R. The Physical Properties of Amorphous Polymer Networks[J]. Chemistry & Industry, 1951 (45):955-958.
- [37] Treloar L R. The Thermodynamic Study of Rubber-Like Elasticity[J]. Proceedings of the Royal Society B-Biological Sciences, 1952, 139(897):506-512.
- [38] Wichterle O, Lim D. Hydrophilic Gels for Biological Use[J]. Nature, 1960, 185(4706):117-118.
- [39] Bell C L, Peppas N A. Measurement of the Swelling Force in Ionic Polymer Networks. III. Swelling Force of Interpolymer Complexes[J]. Journal of Controlled Release, 1995, 37(3):277-280.
- [40] Brannon-Peppas L. Preparation and characterization of crosslinked hydrophilic networks. Absorbent Polymer Technology[J]. Studies in Polymer Sci, 1990 8:46–66.
- [41] Jones D S, Andrews G P, Gorman S P. Characterization of Crosslinking Effects on the Physicochemical and Drug Diffusional Properties of Cationic Hydrogels Designed as Bioactive Urological Biomaterials[J]. Journal of Pharmacy & Pharmacology, 2010, 57(10):1251-1259.
- [42] Ganji F, Vasheghani-Farahani S, Vasheghani-Farahani E. Theoretical Description of Hydrogel Swelling: A Review[J]. Iranian Polymer Journal, 2010, 19(5): p.375-398.
- [43] Mittal H, Ray S S, Okamoto M. Recent Progress on the Design and Applications of Polysaccharide-Based Graft Copolymer Hydrogels as Adsorbents for Wastewater Purification[J]. Macromolecular Materials and Engineering, 2016, 301(5):496-522.
- [44] Zhu J, Marchant R E. Design Properties of Hydrogel Tissue-Engineering Scaffolds[J]. Expert Review of Medical Devices, 2011, 8(5):607-626.
- [45] Maitra J, Shukla VK. Cross-Linking in Hydrogels—A Review[C]. Amer J Polym Sci, 2014, 4:25–31.
- [46] Djabourov M, Leblond J, Papon P. Gelation of Aqueous Gelatin Solutions. I. Structural Investigation[J]. Journal de Physique, 1988, 49(2):319-332.
- [47] Djabourov M, Leblond J, Papon P. Gelation of Aqueous Gelatin Solutions. II. Rheology of the Sol-Gel Transition[J]. Journal de Physique. 1988, 49(2):333-343.
- [48] Lutz J F, Akdemir Ö, Hoth A. Point by Point Comparison of Two Thermosensitive Polymers Exhibiting a Similar LCST: Is the Age of Poly (NIPAM) Over? [J]. Journal of the American Chemical Society, 2006 128(40):13046-13047.
- [49] Zhang S. Fabrication of Novel Biomaterials through Molecular Self-Assembly[J]. Nature Biotechnology, 2003, 21(10):1171-1178.

- [50] O'leary L E, Fallas J A, Bakota E L, Kang M K, Hartgerink J D. Multi-Hierarchical Self-Assembly of a Collagen Mimetic Peptide from Triple Helix to Nanofibre and Hydrogel[J]. *Nature Chemistry*, 2011, 3(10):821-828.
- [51] Braccini I, Pérez S. Molecular Basis of Ca²⁺-Induced Gelation in Alginates and Pectins: The Egg-Box Model Revisited[J]. *Biomacromolecules*, 2001, 2(4):1089-1096.
- [52] Morton S W, Herlihy K P, et al. Scalable Manufacture of Built - to - Order Nanomedicine: Spray - Assisted Layer - By - Layer Functionalization of Print Nanoparticles[J]. *Advanced Materials*, 2013, 25(34):4707-4713.
- [53] Bekturov E A, Bimendina L A, et al. Interpolymer complexes. In *Speciality Polymers*[M]. Springer, 1981, 99-147.
- [54] Hennink WE, van Nostrum C F. Novel Crosslinking Methods to Design Hydrogels[J]. *Advanced Drug Delivery Reviews*, 2012, 64:223-236.
- [55] Li J, Illeperuma W R K, Suo Z, et al. Hybrid Hydrogels with Extremely High Stiffness and Toughness[J]. *Acs Macro Letters*, 2014, 3(6):520-523.
- [56] Annabi, Nasim, et al. Highly Elastic and Conductive Human-Based Protein Hybrid Hydrogels[J]. *Advanced Materials*, 2016, 28(1):40-49.
- [57] Silva S S, Mano J F, Reis R L. Potential Applications of Natural Origin Polymer-Based Systems in Soft Tissue Regeneration[J]. *Critical Reviews in Biotechnology*, 2010, 30(3):200-221.
- [58] Li F, Li S, El Ghzaoui A, Nouailhas H, Zhuo R. Synthesis and Gelation Properties of PEG-PLA-PEG Triblock Copolymers Obtained by Coupling Monohydroxylated PEG-PLA with Adipoyl Chloride[J]. *Langmuir*, 2007, 23(5):2778-2783.
- [59] Wang C, Stewart RJ, Kopeček J. Hybrid Hydrogels Assembled from Synthetic Polymers and Coiled-Coil Protein Domains[J]. *Nature*, 2015, 397(6718):417.
- [60] Lim J, Chouai A, Lo S T, Liu W, Sun X, Simanek E E. Design, Synthesis, Characterization, and Biological Evaluation of Triazine Dendrimers Bearing Paclitaxel Using Ester and Ester/Disulfide Linkages[J]. *Bioconjugate Chemistry*, 2009, 20(11):2154-2161.
- [61] Mitumata T, Suemitsu Y, et al. pH-Response of Chitosan, K-Carrageenan, Carboxymethyl Cellulose Sodium Salt Complex Hydrogels[J]. *Polymer*, 2003, 44(23):7103-7111.
- [62] Jaiswal M, Koul V. Assessment of Multicomponent Hydrogel Scaffolds of Poly(Acrylic Acid-2-Hydroxy Ethyl Methacrylate)/Gelatin for Tissue Engineering Applications[J]. *Journal of Biomaterials Applications*, 2013, 27(7):848-861.
- [63] Gupta K, Singhal R. A Review: Tailor-made Hydrogel Structures (Classifications and Synthesis Parameters[J]. *Journal of Macromolecular Science Part D Reviews*

- in Polymer Processing, 2016, 55(1):54-70.
- [64] Mastropietro D J, Omidian H, Park K. Drug Delivery Applications for Superporous Hydrogels[J]. expert opinion on drug delivery, 2012, 9(1):71-89.
 - [65] Ahmed, Enas M. Hydrogel: Preparation, Characterization, and Applications: A Review[J]. Journal of Advanced Research, 2015, 6(2):105-121.
 - [66] Gupta K, Singhal R. A Review: Tailor-made Hydrogel Structures (Classifications and Synthesis Parameters)[J]. Journal of Macromolecular Science Part D Reviews in Polymer Processing, 2016, 55(1):54-70.
 - [67] Li Q, Xu Z, Ji S, Lv P, Li X, Hong W. Kinetics-Induced Morphing of 3D-Printed Gel Structures Based on Geometric Asymmetry[J]. Journal of Applied Mechanics, 2020 22(7):p.655-670.
 - [68] Tasdelen B, Cifci DI, Meric S. Thermo and pH-Responsive Polymers and Adsorbents for Separation Purposes[M]. Jacobs J Civil Eng, 2015 1:1–3.
 - [69] Buchholz F L, Graham A T. Modern Superabsorbent Polymer Technology[J]. Modern Superabsorbent Polymer Technology - Research and Markets, 1997, Vol. 152.
 - [70] Saraydin D, Karadağ E, Oztop H N, et al. Adsorption of Bovine Serum Albumin onto Acrylamide-Maleic Acid Hydrogels[J]. Biomaterials, 1994, 15(11):917.
 - [71] Karada E, Saraydin D, Olgun Güven. Interaction of Nicotine and Its Pharmaceutical Derivatives with Acrylamide/Itaconic Acid Hydrogels[J]. Journal of Applied Polymer ence, 1997, 66(4).
 - [72] Chujo Y, Sada K, Saegusa T. Cobalt(III) Bipyridyl-Branched Polyoxazoline Complex as a Thermally and Redox Reversible Hydrogel[J]. Macromolecules, 1993, 26(24):6320-6323.
 - [73] Karadağ E, Saraydin D, Güven O. Behaviors of Acrylamide/Itaconic Acid Hydrogels in Uptake of Uranyl Ions from Aqueous Solutions[J]. Separation Science and Technology, 1995, 30(20):3747-3760.
 - [74] Akkas P, Olgun Güven. Enhancement of Uranyl Ion Uptake by Prestructuring of Acrylamide-Maleic Acid Hydrogels[J]. Journal of Applied Polymer Science, 2000, 78(2):284-289.
 - [75] Saraydin D, Karadağ E, Güven O. Adsorption of Some Basic Dyes by Acrylamide-Maleic Acid Hydrogels[J]. Separation Science & Technology, 1996, 31(3):423-434.
 - [76] Kloxin A M, Kloxin C J, Bowman C N, et al. Mechanical Properties of Cellularly Responsive Hydrogels and Their Experimental Determination[J]. advanced materials, 2010, 22(31):3484-3494.
 - [77] Wang Y, Chen L, Tan L, Zhao Q, Luo F, Wei Y, Qian Z. PEG-PCL Based Micelle Hydrogels as Oral Docetaxel Delivery Systems for Breast Cancer Therapy[J]. Biomaterials, 2014, 35(25):6972-6985.

- [78] Lee K Y, Mooney D J. Alginate: Properties and Biomedical Applications[J]. Progress in polymer science, 2012, 37(1):106-126.
- [79] Hu X, Gong X. A New Route to Fabricate Biocompatible Hydrogels with Controlled Drug Delivery Behavior[J]. Journal of Colloid & Interface Science, 2016, 470:62-70.
- [80] Das D, Ghosh P, Dhara S, et al. Dextrin and Poly(acrylic acid)-Based Biodegradable, Non-Cytotoxic, Chemically Cross-Linked Hydrogel for Sustained Release of Ornidazole and Ciprofloxacin[J]. Acs Applied Materials & Interfaces, 2015, 7(8):4791-4803.
- [81] Caló E, Khutoryanskiy V. Biomedical Applications of Hydrogels: A Review of Patents and Commercial Products[J]. European Polymer Journal, 2015, 65:252-267.
- [82] Stashak T S, Farstvedt E, Othie A. Update on Wound Dressings: Indications and Best Use[J]. Clinical Techniques in Equine Practice, 2004, 3(2):148-163.
- [83] Wang J, Wang L, Yu H, et al. Recent Progress on Synthesis, Property and Application of Modified Chitosan: An Overview[J]. International Journal of Biological Macromolecules, 2016, 88:333-344.
- [84] Drury J L, Mooney D J. Hydrogels for Tissue Engineering: Scaffold Design Variables and Applications[J]. Biomaterials, 2003, 24(24):4337-4351.
- [85] Li Y, Rodrigues J, Tomás, Helena. Injectable and Biodegradable Hydrogels: Gelation, Biodegradation and Biomedical Applications[J]. Chemical Society Reviews, 2012, 41(6):2193-2221.
- [86] Lee K Y, Mooney D J. Hydrogels for Tissue Engineering[J]. 101(7):1869-1880.
- [87] Romero M R, Wolfel A, Igarzabal C I. Smart Valve: Polymer Actuator to Moisture Soil Control[J]. Sensors and Actuators B Chemical, 2016, 234:53-62.
- [88] Gong J P, Katsuyama Y, Kurokawa T, et al. Double-Network Hydrogels with Extremely High Mechanical Strength[J]. Advanced materials, 2003, 15(14):1155-1158.
- [89] Oh J Y, Rondeau-Gagné S, Chiu Y C, et al. Intrinsically Stretchable and Healable Semiconducting Polymer for Organic Transistors[J]. Nature, 2016, 539(7629):411-415.
- [90] Zhang Y N, Avery R K, et al. A Highly Elastic and Rapidly Crosslinkable Elastin-Like Polypeptide-Based Hydrogel for Biomedical Applications[J]. Advanced Functional Materials, 2015, 25(30):4814-4826.
- [91] Li J, Suo Z, Vlassak J J. Stiff, Strong, and Tough Hydrogels with Good Chemical Stability[J]. Journal of Materials Chemistry B, 2014, 2(39):6708-6713.
- [92] Haraguchi K, Takehisa T. Nanocomposite Hydrogels: A Unique Organic-Inorganic Network Structure with Extraordinary Mechanical, Optical, and Swelling/De-swelling Properties[J]. Advanced Materials, 2002, 14(16).

- [93] Yuk H, Zhang T, Parada GA, Liu X, Zhao X. Skin-Inspired Hydrogel-Elastomer Hybrids with Robust Interfaces and Functional Microstructures[C]. Nat. Commun. 7,2016, 12028.
- [94] Avery R K, Albadawi H, Akbari M, et al. An Injectable Shear-Thinning Biomaterial for Endovascular Embolization[J]. Science translational medicine, 2016, 8(365):365ra156-365ra156.
- [95] Paul A, Hasan A, Kindi H A, et al. Injectable Graphene Oxide/Hydrogel-Based Angiogenic Gene Delivery System for Vasculogenesis and Cardiac Repair[J]. Acs Nano, 2014, 8(8):8050-8062.
- [96] Phadke A, Zhang C, Arman B, et al. Rapid Self-Healing Hydrogels[J]. Proceedings of the National Academy of Sciences, 2012, 109(12):p.4383-4388.
- [97] Tibbitt M W, Anseth K S. Dynamic Microenvironments: The Fourth Dimension[J]. Science Translational Medicine, 2012, 4(160):160ps24-160ps24.
- [98] Zhang L, Liang H, Jacob J, et al. Photogated Humidity-Driven Motility[J]. Nature Communications, 2015, 6:7429.
- [99] Breger J C, Yoon C K, Xiao R, et al. Self-Folding Thermo-Magnetically Responsive Soft Microgrippers[J]. Acs Applied Materials & Interfaces, 2015, 7(5):3398-3405.
- [100] Shin SR, Jung SM, Zalabany M, et al. Carbon-Nanotube-Embedded Hydrogel Sheets for Engineering Cardiac Constructs and Bioactuators[J]. Acs Nano, 2013, 7(3):2369-2380.
- [101] Du Y, Lo E, Ali S, et al. Directed Assembly of Cell-Laden Microgels for Fabrication of 3D Tissue Constructs[J]. Proceedings of the National Academy of Sciences of the United States of America, 2008, 105(28):9465-9470.
- [102] Qi H, Ghodousi M, Du Y, et al. DNA Directed Self-Assembly of Shape-Controlled Hydrogels[J]. Nature Communications, 2013, 4:2275.
- [103] Colosi C, Shin SR, Manoharan V, Massa S, et al. Microfluidic Bioprinting of Heterogeneous 3D Tissue Constructs Using Low-Viscosity Bioink[J]. Advanced Materials, 2016 28(4):677-684.
- [104] Kamata H, Akagi Y, Kayasuga-Kariya Y, et al. 'Nonswellable' Hydrogel Without Mechanical Hysteresis[J]. Science, 2014, 343(6173):873-875.
- [105] Hong S, Sycks D, Chan HF, et al. 3D Printing of Highly Stretchable and Tough Hydrogels into Complex, Cellularized Structures[J]. Advanced Materials, 2015 27, 4035–4040.
- [106] Flory P J. Thermodynamics of High Polymer Solutions[J]. The Journal of Chemical Physics, 1942, 10 (1), 51–61.
- [107] Wang X, Hong W, et al. A Visco-Poroelastic Theory for Polymeric Gels[J]. Proceedings Mathematical Physical & Engineering Sciences, 2012, 468(2148):3824-3841.

- [108] Gedde U L. Polymer Physics[M], Oxford University Press, New York, 2003.
- [109] Long R, Mayumi K, Creton C, et al. Time Dependent Behavior of a Dual Cross-Link Self-Healing Gel: Theory and Experiments[J]. *Macromolecules*, 2014, 47(20):7243-7250.
- [110] Mayumi K, Marcellan A, Ducouret G, et al. Stress–Strain Relationship of Highly Stretchable Dual Cross-Link Gels: Separability of Strain and Time Effect[J]. *ACS Macro Letters*, 2013 2(12):1065-1068.
- [111] Guo J, Long R, Mayumi K, et al. Mechanics of a Dual Cross-Link Gel with Dynamic Bonds: Steady State Kinetics and Large Deformation Effects[J]. *Macromolecules*, 2016, 49(9):3497-3507.
- [112] Guo J, Liu M, Zehnder A T, et al. Fracture Mechanics of a Self-Healing Hydrogel with Covalent and Physical Crosslinks: a Numerical Study[J]. *Journal of the Mechanics and Physics of Solids*, 2018, 120:79-95.
- [113] Chen C M, Yang S. Wrinkling Instabilities in Polymer Films and Their Applications[J]. *polymer international*, 2012, 61(7):1041-1047.
- [114] Logg A, Wells GN. DOLFIN: Automated Finite Element Computing[J]. *ACM Transactions on Mathematical Software (TOMS)*, 2010, 37(2):1-28.
- [115] Logg A, Wells GN, Hake J. DOLFIN: A C++/Python Finite Element Library. In *Automated Solution of Differential Equations by the Finite Element Method*[M]. Springer, Berlin, Heidelberg, 2012, 173-225.
- [116] Kirby R C, Logg A. A Compiler for Variational Forms[J]. *ACM Transactions on Mathematical Software (TOMS)*, 2006, 32(3):417-444.
- [117] Logg A, Ølgaard K B, Rognes M E, Wells G N. FFC: the Fenics Form Compiler. In *Automated Solution of Differential Equations by the Finite Element Method*[M]. Springer, Berlin, Heidelberg, 2012, 227-238.
- [118] Ølgaard K B, Wells G N. Optimizations for Quadrature Representations of Finite Element Tensors through Automated Code Generation[J]. *ACM Transactions on Mathematical Software (TOMS)*, 2010, 37(1):1-23.
- [119] Alnæs M S, Logg A, Ølgaard K B, Rognes M E, Wells G N. Unified Form Language: A Domain-Specific Language for Weak Formulations of Partial Differential Equations[J]. *ACM Transactions on Mathematical Software (TOMS)*, 2014, 40(2):1-37.
- [120] Alnæs M S. UFL: A Finite Element Form Language. In *Automated Solution of Differential Equations by the Finite Element Method*[M]. Springer, Berlin, Heidelberg, 2012, 40(2):1-37.
- [121] Kirby R C. Algorithm 839: FIAT, A New Paradigm for Computing Finite Element Basis Functions[J]. *ACM Transactions on Mathematical Software (TOMS)*, 2004, 30(4):502-516.
- [122] Kirby R C. FIAT: Numerical Construction of Finite Element Basis Functions.

- Inautomated Solution of Differential Equations by the Finite Element Method[M]. Springer, Berlin, Heidelberg, 2012, 247-255.
- [123] Alnæs M S, Logg A, Mardal K A, Skavhaug O, Langtangen H P. Unified framework for finite element assembly[J]. International Journal of Computational Science and Engineering, 2009, 4(4):231-244.
- [124] Alnæs M S, Logg A, Mardal K A. UFC: A Finite Element Code Generation Interface. Inautomated Solution of Differential Equations by the Finite Element Method[M]. Springer, Berlin, Heidelberg, 2012, 283-302.
- [125] Sun J, Zhao X, et al. Highly Stretchable and Tough Hydrogels[J]. Nature, 2012, 489(7414):133-136.

附录 A 文中 FEniCS 计算涉及到的 Python 源代码

在文章的章节 4.1 介绍了 FEniCS 的组件及各个组件对应的功能，下面简单介绍一下 FEniCS 的常用对象、函数定义以及求解器设置，供答辩委员或读者参考。更加详细的功能及案例介绍请参考官方教程，附赠下载地址 <https://book.cc/2035653/834e6a>。

A.1 FEniCS 程序中变量与正文中公式对应变量对照表

表 A-1 COMSOL 模型的关键参数

FEniCS 程序中对象	公式变量	说明
V	\	矢量场
Q	\	标量场
u	u	位移
p	Π	拉格朗日乘子
v	v	位移试函数
q	$\delta\Pi$	拉格朗日乘子试函数
wt(v, mu_)	\	试函数
dw	\	测试函数
F	F	变形梯度
B	B	体力
T	T	面力
C	\	右柯西格林张量
J	$\det F$	$\det(F)$
E	\	杨氏模量
nu	\	泊松比
psi	W_s	弹性能密度
tr	tr()	求迹
a	\	弱形式
J1	\	雅克比矩阵
*dx	* dV	体积分（三维）/面积分（二维）
*ds	* dS	面积分（三维）/线积分（二维）

表 A-1 (续)

FEniCS 程序中对象	公式变量	说明
C0	\backslash	初始时水的相对体积
lambda0	λ_0	预溶胀拉伸比
Nv	Nv	无量纲弹性模量
chi	χ	混合焓系数
mu0	0.01	初始时刻化学势
vis	η	无量纲动力粘度
mu	μ	化学势场
mu_D	\backslash	化学势边界条件
Ic	I_1	右柯西格林张量第一主不变量
Ic2	I_2	右柯西格林张量第二主不变量
Fv	$\delta \mathbf{F}$	变形梯度的变分
Cs	C	水的相对体积
gradv	$\frac{\partial v}{\partial t}$	速度的时间导数
W_FC	$W(F,C)$	自由能密度
W_vis	W_{vis}	粘度耗散项
v_const	v	标度指数(0.588)
phi0v	ϕ_0	初始时刻高分子体积分数
u_n	u_n	上一时刻位移(差分法)
mu_n	μ_n	上一时刻化学势(差分法)
w_n	\backslash	上一时刻混合场 (差分法)
H	H	变形梯度的逆
phi	ϕ	高分子体积份数
lambda_ref	λ_{ref}	高分子长链波动拉伸比
Pi0		势函数
Wrinkle 模型		
C0_0	\backslash	基底中初始时水的相对体积
C0_1	\backslash	薄膜中初始时水的相对体积
Nv_0	\backslash	基底无量纲弹性模量
Nv_1	\backslash	薄膜无量纲弹性模量

表 A-1 (续)

FEniCS 程序中对象	公式变量	说明
lambda01	\	基底预溶胀拉伸比
lambda02	\	薄膜预溶胀拉伸比
mu_0	\	基底初始时刻化学势
mu_1	\	薄膜初始时刻化学势
混合单元模型		
Pi	Π	拉格朗日乘子
Cs	C	水的相对体积
Pi_n	\	上一时刻拉格朗日乘子(差分法)
Cs_n	\	上一时刻水的相对体积(差分法)
Ct	$\frac{\partial C}{\partial t}$	水的相对体积时间导数
两种溶剂的模型		
C10	\	初始时刻溶剂 1 的相对体积
C20	\	初始时刻溶剂 2 的相对体积
C1_final	\	最终溶剂 1 的相对体积
C2_final	\	最终溶剂 2 的相对体积
J0	J_0	初始时刻总体积
lambda0	λ_0	初始时刻预溶胀拉伸比
J_final	J_{final}	最终总体积
lambda_final	λ_{final}	最终溶胀拉伸比
chi1p	χ_{1p}	溶剂 1 与高分子的混合焓系数
chi2p	χ_{2p}	溶剂 2 与高分子的混合焓系数
chi12	χ_{12}	溶剂 1 与溶剂 2 的混合焓系数
mu10	\	初始时刻溶剂 1 化学势
mu20	\	初始时刻溶剂 2 化学势
mu1_final	\	最终溶剂 1 化学势
mu2_final	\	最终溶剂 2 化学势
D1	D_1	溶剂 1 扩散系数
D2	D_2	溶剂 2 扩散系数
Cs1	C_1	溶剂 1 相对体积场
Cs2	C_2	溶剂 2 相对体积场

表 A-1 (续)

FEniCS 程序中对象	公式变量	说明
mu1	μ_1	溶剂 1 化学势场
mu2	μ_2	溶剂 2 化学势场
wt(v, Pi_, Cs1_, mu1_, Cs2_, mu2_)	\	试函数
w_n(u_n, Pi_n, Cs1_n, mu1_n, Cs2_n, mu2_n)	\	上一时刻的变量场(差分法)
C1t	$\frac{\partial C_1}{\partial t}$	溶剂 1 的时间导数
C2t	$\frac{\partial C_2}{\partial t}$	溶剂 2 的时间导数

A.2 FEniCS 常用 Python 函数对象

UnitSquare(nx, ny): 用于定义大小为 $[0,1] \times [0,1]$ 的矩形的网格，nx、ny 分别为 x、y 方向上划分的网格数量，网格类型为三角形网格。同类型的函数对象还包括 UnitInterval, UnitCube, UnitCircle, UnitSphere, Interval, Rectangle, Box 等。使用方法如下：

```

1.  # 1D domains
2.  mesh = UnitInterval(20) # 20 cells, 21 vertices
3.  mesh = Interval(20, -1, 1) # domain [-1,1]
4.  # 2D domains (6x10 divisions, 120 cells, 77 vertices)
5.  mesh = UnitSquare(6, 10) # "right" diagonal is default
6.  # The diagonals can be right, left or crossed
7.  mesh = UnitSquare(6, 10, "left")
8.  mesh = UnitSquare(6, 10, "crossed")
9.  # Domain [0,3]x[0,2] with 6x10 divisions and left diagonals
10. mesh = Rectangle(0, 0, 3, 2, 6, 10, "left")
11. # 6x10x5 boxes in the unit cube, each box gets 6 tetrahedra:
12. mesh = UnitCube(6, 10, 5)
13. # Domain [-1,1]x[-1,0]x[-1,2] with 6x10x5 divisions
14. mesh = Box(-1, -1, -1, 1, 0, 2, 6, 10, 5)
15. # 10 divisions in radial directions
16. mesh = UnitCircle(10)
    
```

```
17. mesh = UnitSphere(10)
```

FEniCS 支持外部导入网格文件，文件是后缀为.xml 或者.xml.gz 格式的文本文件，文件的格式为如下所示：

```
1.  <?xml version="1.0" encoding="UTF-8"?>
2.
3.  <dolfin xmlns:dolfin="http://fenicsproject.org">
4.    <mesh celltype="tetrahedron" dim="3">
5.      <vertices size="624">
6.        <vertex index="0" x="0.0" y="0.0" z="0.0" />
7.        <vertex index="1" x="0.0" y="0.0" z="4.0" />
8.        ...
9.      </vertices>
10.     <cells size="2217">
11.       <tetrahedron index="0" v0="432" v1="464" v2="465" v3="433" />
12.       <tetrahedron index="1" v0="465" v1="466" v2="467" v3="464" />
13.       ...
14.     </cells>
15.   </mesh>
16. </dolfin>
```

在上述.xml 文件中给定几何的维度，单元类型，节点信息以及单元信息即可。可以通过在 ANSYS、Abaqus 中快速定义好需要使用的网格，从对应的工程文件中读取相关的节点以及单元信息，通过简单的编程即可得到相应的.xml 文件。下面给出笔者的 Python 函数：

```
1.  import numpy as np
2.  import pandas as pd
3.  from sklearn.model_selection import ShuffleSplit
4.  import csv
5.  import matplotlib.pyplot as plt
6.  from pandas import DataFrame
7.  from scipy.linalg import solve
8.  from scipy.interpolate import griddata
```



```

9.     import matplotlib as mpl
10.    n_nodes = 624
11.    n_elements = 2217
12.
13.    # read data
14.    f=open('E:\Job-1.txt','r')
15.    n=0
16.    Data={} #记录文件所有行数据信息的字典，检索方式为行数
17.    Num={} #记录关键字所在行数的字典，检索方式为关键字字符串
18.    for line in f:
19.        Data[n]=l = list(map(str, line.split())) #将整个 line 分隔
20.        n=n+1
21.
22.    #定义数组储存应变应力信息
23.    linenum=0
24.    nodes = np.zeros(shape=(n_nodes,4))
25.    elements = np.zeros(shape=(n_elements,5),dtype = int)
26.    #从文件中获取单元和节点信息
27.    for i in range(n_nodes):
28.        for j in range(4):
29.            nodes[i,j] = float(Data[i][j])
30.    print(Data[n_nodes])
31.    for i in range(n_elements):
32.        for j in range(5):
33.            elements[i,j] = int(Data[n_nodes+i][j]) - 1
34.    f.close()
35.
36.    with open("E:\\two_solvents0415.txt","w") as f:
37.        f.write('<?xml version="1.0" encoding="UTF-8"?>')
38.        f.write('\n')
39.        f.write('\n')
40.        f.write('<dolfin xmlns:dolfin="http://fenicsproject.org">')

```

```

41.     f.write('\n')
42.     f.write(' <mesh celltype="tetrahedron" dim="3">')
43.     f.write('\n')
44.     f.write(' <vertices size="')
45.     f.write(str(n_nodes))
46.     f.write('">')
47.     f.write('\n')
48.     for i in range(n_nodes): #对于双层列表中的数据
49.         nnn = str(' ') + str('<vertex index="') + str(i) + str('" x="') + str(nodes[i,1]) \
50.             + str('" y="') + str(nodes[i,2]) + str('" z="') + str(nodes[i,3]) + str('" />')
51.         f.write(nnn)
52.         f.write('\n')
53.     f.write(' </vertices>')
54.     f.write('\n')
55.     f.write(' <cells size="')
56.     f.write(str(n_elements))
57.     f.write('">')
58.     f.write('\n')
59.     for i in range(n_elements):
60.         nnn = str(' ') + str('<tetrahedron index="') + str(i) + str('" v0="') \
61.             + str(elements[i,1]) + str('" v1="') + str(elements[i,2]) + str('" v2="') \
62.             + str(elements[i,3]) + str('" v3="') + str(elements[i,4]) + str('" />')
63.         f.write(nnn)
64.         f.write('\n')
65.     f.write(' </cells>')
66.     f.write('\n')
67.     f.write(' </mesh>')
68.     f.write('\n')
69.     f.write('</dolphin>')
70.     f.write('\n')
71.     f.close

```

FunctionSpace(mesh, element_type, degree): 在网格上定义的函数空间, 具有给定的元素类型(如“Lagrange”或“DG”)(“DG”是 Discontinuous Galerkin); degree 为单元阶数。对于多场耦合问题的多重函数空间可以通过以下方法来定义:

```

1.   V = VectorElement('P',tetrahedron,1)
2.   Q1 = FiniteElement('P',tetrahedron,1)
3.   Q2 = FiniteElement('P',tetrahedron,1)
4.   MU1 = FiniteElement('P',tetrahedron,1)
5.   MU2 = FiniteElement('P',tetrahedron,1)
6.   P = FiniteElement('P',tetrahedron,1)
7.   J_P = FiniteElement('P',tetrahedron,1)
8.   TH = MixedElement([V,Q1,Q2,MU1,MU2,P])
9.   W = FunctionSpace(mesh,TH)
10.  #fixed left and bottom sides and Define chemical potential boundary conditions
11.  bcl1 = DirichletBC(W.sub(0).sub(0), Constant(0), left)
12.  bcl2 = DirichletBC(W.sub(0).sub(2), Constant(0), left)
13.  bcf = DirichletBC(W.sub(0).sub(1), Constant(0), front)
14.  bcb_C1 = DirichletBC(W.sub(1), C1_bc_RU, bottom)
15.  bcb_C2 = DirichletBC(W.sub(2), C2_bc_RU, bottom)
16.  bcb_mu1 = DirichletBC(W.sub(3), mu1_bc_RU, bottom)
17.  bcb_mu2 = DirichletBC(W.sub(4), mu2_bc_RU, bottom)
18.  bcs = [bcl1,bcl2,bcf,bcb_C1, bcb_C2, bcb_mu1,bcb_mu2]
```

Expression(formula, p1=v1, p2=v2, ...): 用来定义一个标量或向量值函数, 用 c++ 语法写成数学表达式。表达式中的空间坐标分别为 x[0]、x[1]、x[2], 表达式中的常用的数学计算符包括: cos, sin, tan, acos, asin, atan, atan2, cosh, sinh, tanh, exp, frexp, ldexp, log, log10, modf, pow, sqrt, ceil, fabs, floor, 和 fmod 等。该对象中定义的 p1、p2 可以通过类似 Expression.p1=... 进行更新。例如:

```

1.   mu_Dbc = Expression("mu00+(-0.221-mu00)*
((x[0]+x[1])*(x[0]+x[1])+1)*pow((t),0.5)", degree = 1, mu00 = mu10, t = 0)
```

Function(V): 用来定义函数空间 V 对应的函数空间对象。

SubDomain: 用于定义子域类, 可以是边界的一部分、内部边界, 也可以是

域的一部分。

Mesh:用于定义有限元网格的类, 包含 **cells**, **vertices**, optionally **faces**, **edges** 和 **facets**。

MeshFunction:标记区域或边界部分的工具。

DirichletBC(V, value, where):用于给定对应的函数空间 **V** 的 **Dirichlet** 边界条件, **value** 可以是常数值, 也可以是 **Expression** 函数; **where** 可以是边界, 也可以是 **SubDomain**, **MeshFunction** 对象。

TrialFunction(V): 用于定义一个函数空间 **V** 上的试验函数, 用变分形式表示有限元问题中的未知项。

TestFunction(V): 用于定义一个函数空间 **V** 上的试函数, 用在控制方程中的变分形式。

LinearVariationalProblem(a, L, u, bcs): 定义一个变分问题, 给出一个双线性项 (**a**)和线性项(**L**), 使用 **UFL** 语法编写, 指定需要求解的函数空间对象以及边界条件。

LinearVariationalSolver(problem): 为一个线性变分问题对象创建求解器对象。

plot(q):用于 **mesh**, **Function**, **MeshFunction** 对象以及 **Function** 对象函数的快速可视化。

interpolate(func, V): 用于将一个公式或有限元函数插值到函数空间 **V** 中。

project(func, V): 用于将一个公式或有限元函数映射到函数空间 **V** 上。

derivative(Pi0, w, wt): 用于求变分。

设置求解器参数:

```
parameters["form_compiler"]["cpp_optimize"] = True
parameters["form_compiler"]["representation"] = "uflacs"
parameters["form_compiler"]["quadrature_degree"] = 3
parameters={"newton_solver":{"relative_tolerance":1e-4, "absolute_tolerance":1e-6, "maximum_iterations":20}}
solve(a0==0, w, bcs, J = J0, solver_parameters=parameters)
```

A.3 FEniCS 常用求解器以及预处理

FEniCS 常用的求解器如下表 A-2 所示:

表 A-2 FEniCS 常用的求解器

名称	方法
"lu"	sparse LU factorization (Gaussian elim.)
"cholesky"	sparse Cholesky factorization
"cg"	Conjugate gradient method
"gmres"	Generalized minimal residual method
"bicgstab"	Biconjugate gradient stabilized method
"minres"	Minimal residual method
"bicgstab"	Biconjugate gradient stabilized method
"minres"	Minimal residual method
"tfqmr"	Transpose-free quasi-minimal residual method
"richardson"	Richardson method

FEniCS 常用的预处理方法如下表 A-3 所示:

表 A-3 FEniCS 常用的预处理方法

名称	方法
"none"	No preconditioner
"ilu"	Incomplete LU factorization
"icc"	Incomplete Cholesky factorization
"jacobi"	Jacobi iteration
"bjacobi"	Block Jacobi iteration
"sor"	Successive over-relaxation
"amg"	Algebraic multigrid (BoomerAMG or ML)
"additive_schwarz"	Additive Schwarz
"hypre_amg"	Hypre algebraic multigrid (BoomerAMG)
"hypre_euclid"	Hypre parallel incomplete LU factorization
"hypre_parasails"	Hypre parallel sparse approximate inverse
"ml_amg"	ML algebraic multigrid

使用方法如下所示：

```
1.   prm = LinearVariationalSolver.default_parameters()
2.   if linear_solver == 'Krylov':
3.       prm.linear_solver = 'gmres'
4.       prm.preconditioner = 'ilu'
5.       prm.krylov_solver.absolute_tolerance = abs_tol
6.       prm.krylov_solver.relative_tolerance = rel_tol
7.       prm.krylov_solver.maximum_iterations = max_iter
8.   else:
9.       prm.linear_solver = 'lu'
10.  u = Function(V)
11.  solve(a == L, u, bcs, solver_parameters=prm)
```

附录 B 文中 FEniCS 计算涉及到的 Python 源代码

在文章中，部分章节展示了 FEniCS 有限元程序的部分内容。在附录的该部分将给出文中计算水凝胶溶胀各个模型的所涉及到的源代码，供答辩委员或读者参考。

B.1 FEniCS 二维化学凝胶溶胀模型

```

1.  from __future__ import print_function
2.  import matplotlib.pyplot as plt
3.  from fenics import *
4.  import numpy as np
5.  # Optimization options for the form compiler
6.  parameters["form_compiler"]["cpp_optimize"] = True
7.  parameters["form_compiler"]["representation"] = "uflacs"
8.  parameters["form_compiler"]["quadrature_degree"] = 3
9.  # Create mesh and define function space
10. nx = 40
11. ny = 40
12. mesh = RectangleMesh(Point(0.0,0.0),Point(20.0,20.0),nx,ny)
13. #define global parameters and initial parameters
14. C0 = 0.2
15. lambda0 = (1+C0)**(1/3)
16. Nv = 0.001
17. chi = 0.2
18. mu0 = ln(C0/(1+C0))+1/(1+C0)+chi/(1+C0)**2+Nv*(1/lambda0-\
19.         1/(1+C0)) # mu0 = -0.81943
20. vis = 0.01
21. # Create mesh and define function space
22. V = VectorElement('P',triangle,2)
23. Q = FiniteElement('P',triangle,1)
24. TH = V*Q
25. V1 = FunctionSpace(mesh, V)

```

```

26. Q1 = FunctionSpace(mesh, Q)
27. W = FunctionSpace(mesh, TH)
28. # Mark boundary subdomians
29. left = CompiledSubDomain("near(x[0], side) && on_boundary", side = 0.0)
30. right = CompiledSubDomain("near(x[0], side) && on_boundary", side = 20.0)
31. top = CompiledSubDomain("near(x[1], side) && on_boundary", side = 20.0)
32. bottom = CompiledSubDomain("near(x[1], side) && on_boundary", side = 0.0)
33. # Define chemical potential boundary conditions
34. mu_D = Expression("mu00*(1-t/100.0)", degree = 1, mu00 = mu0, t = 0)
35. #fixed left and bottom sides and Define chemical potential boundary conditions
36. bcl = DirichletBC(W.sub(0).sub(0), Constant(0), left)
37. bcb = DirichletBC(W.sub(0).sub(1), Constant(0), bottom)
38. bct_mu = DirichletBC(W.sub(1), mu_D, top)
39. bcr_mu = DirichletBC(W.sub(1), mu_D, right)
40. bcs = [bcl, bcb, bct_mu, bcr_mu]
41. # Define test functions
42. w = Function(W)
43. u, mu = split(w)
44. wt = TestFunction(W)
45. v, mu_ = split(wt)
46. dw = TrialFunction(W)
47. w_n = Function(W)
48. u_n, mu_n = split(w_n)
49. # Define deformation gradient tensor
50. F = grad(u) + Identity(len(u))
51. I_ = Identity(len(u))
52. Fv = grad(v)
53. F_mu = grad(mu)
54. F_muv = grad(mu_)
55. H = inv(F)
56. C = F.T*F # Right Cauchy-Green tensor
57. C_2 = C*C

```



```

58. # Invariants of deformation tensors
59. Ic = tr(C)
60. Ic2 = 0.5*(tr(C)*tr(C)-tr(C_2))
61. J = det(F)
62. C1 = lambda0**3*det(F)-1
63. # Save solution in VTK format
64. vtkfile_u = File("2D_gel_D1_t1_n5/displacement.pvd");
65. vtkfile_mu = File("2D_gel_D1_t1_n5/chemical_potential.pvd");
66. vtkfile_phi = File("2D_gel_D1_t1_n5/polumer_volume_fraction.pvd");
67. vtkfile_stress = File("2D_gel_D1_t1_n5/cauchy_stress.pvd");
68. #initial chemical potential value in the domian
69. e_u0 = Expression(('0', '0'), degree=1)
70. e_mu0 = Expression('mu0', mu0=mu0, degree=1)
71. assign(w, [interpolate(e_u0, V1), interpolate(e_mu0, Q1)])
72. #define govern equation
73. dt = Expression('dt1', degree=0, dt1=0.1)
74. gradv = grad((u-u_n)/dt)
75. W_vis = vis*det(F)*(0.5*inner(gradv*H, (gradv*H+H.T*gradV.T)) - \
76.             1/3* tr(gradv*H)* tr(gradv*H))
77. W_FC = 0.5*Nv*(lambda0**2*Ic-3-2*ln(lambda0**3*det(F))) - \
78.         C1*(ln(1+1/C1))-chi/(1+C1)
79. Pi0 = -W_FC - W_vis*dt
80. a0 = (derivative(Pi0, w, wt) + mu*derivative(C1, w, wt))*dx + \
81.       (-C1*dot(H.T*grad(mu),H.T*grad(mu_)) - \
82.       lambda0**3*det(F)*inner(gradv,H.T)*mu_)*dx
83. J1 = derivative(a0, w, dw)
84. # Newton solver parameters
85. parameters={"newton_solver":{"relative_tolerance":1e-4, \
86.                               "absolute_tolerance":1e-6, "maximum_iterations":10}}
87. t = 0.0
88. for n in range(10+89+82):
89.     # Update previous solution

```

```

90.     w_n.assign(w)
91.     # Update current time range(0,0.1,1) range(2,1,90) range(90,5,500)
92.     if t < 0.99:
93.         dt.dt1 = 0.1
94.     elif t < 89.99:
95.         dt.dt1 = 1
96.     elif t < 499.9:
97.         dt.dt1 = 5
98.     t += dt.dt1
99.     if t > 89.9:
100.        mu_D.t = 90.0
101.    else:
102.        mu_D.t = t
103.    print(t)
104.
105.    solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
106.
107.    p1 = plot(w.sub(1),title=str(t))
108.    plt.colorbar(p1)
109.    plt.show()
110.    p1 = plot(w.sub(0), mode = "displacement",title = 'displacement t='+str(t))
111.    plt.colorbar(p1)
112.    plt.show()
113.    vtkfile_u << (w.sub(0), t)
114.    vtkfile_mu << (w.sub(1), t)

```

B.2 FEniCS 二维化学凝胶溶胀混合单元模型

```

1.     from __future__ import print_function
2.     import matplotlib.pyplot as plt
3.     from fenics import *
4.     import numpy as np
5.     # Optimization options for the form compiler

```

```

6. parameters["form_compiler"]["cpp_optimize"] = True
7. parameters["form_compiler"]["representation"] = "uflacs"
8. parameters["form_compiler"]["quadrature_degree"] = 3
9. # Create mesh and define function space
10. nx = 40
11. ny = 40
12. mesh = RectangleMesh(Point(0.0,0.0),Point(20.0,20.0),nx,ny)
13. #define global parameters and initial parameters
14. C0 = 0.2
15. lambda0 = (1+C0)**(1/3)
16. Nv = 0.001
17. chi = 0.2
18. mu0 = ln(C0/(1+C0))+1/(1+C0)+chi/(1+C0)**2+Nv*(1/lambda0-\
19.         1/(1+C0)) # mu0 = -0.81943
20. vis = 0.01
21. # Create mesh and define function space
22. V = VectorElement('P',triangle,2)
23. Q = FiniteElement('P',triangle,1)
24. TH = MixedElement([V, Q, Q, Q])
25. V1 = FunctionSpace(mesh, V)
26. Q1 = FunctionSpace(mesh, Q)
27. W = FunctionSpace(mesh,TH)
28. # Mark boundary subdomians
29. left = CompiledSubDomain("near(x[0], side) && on_boundary", side = 0.0)
30. right = CompiledSubDomain("near(x[0], side) && on_boundary", side = 20.0)
31. top = CompiledSubDomain("near(x[1], side) && on_boundary", side = 20.0)
32. bottom = CompiledSubDomain("near(x[1], side) && on_boundary", side = 0.0)
33. # Define chemical potential boundary conditions
34. mu_D = Expression("mu00*(1-t/100.0)", degree = 1, mu00 = mu0, t = 0)
35. #fixed left and bottom sides and Define chemical potential boundary conditions
36. bcl = DirichletBC(W.sub(0).sub(0), Constant(0), left)
37. bcb = DirichletBC(W.sub(0).sub(1), Constant(0), bottom)

```

```

38. bct_mu = DirichletBC(W.sub(3), mu_D,top)
39. bcr_mu = DirichletBC(W.sub(3), mu_D, right)
40. bcs = [bcl,bcb,bct_mu,bcr_mu]
41. # Define test functions
42. w = Function(W)
43. u, Pi, Cs, mu = split(w)
44. wt = TestFunction(W)
45. v, Pi_, Cs_, mu_ = split(wt)
46. dw = TrialFunction(W)
47. w_n = Function(W)
48. u_n, Pi_n, Cs_n, mu_n = split(w_n)
49. # Define deformation gradient tensor
50. F = grad(u) + Identity(len(u))
51. I_ = Identity(len(u))
52. Fv = grad(v)
53. F_mu = grad(mu)
54. F_muv = grad(mu_)
55. H = inv(F)
56. C = F.T*F # Right Cauchy-Green tensor
57. C_2 = C*C
58. # Invariants of deformation tensors
59. Ic = tr(C)
60. Ic2 = 0.5*(tr(C)*tr(C)-tr(C_2))
61. J = det(F)
62. J_v = det(F)*H*Fv
63. C1 = lambda0**3*det(F)-1
64. # Save solution in VTK format
65. vtkfile_u = File("2D_gel_D1_t1_n5/displacement.pvd");
66. vtkfile_mu = File("2D_gel_D1_t1_n5/chemical_potential.pvd");
67. vtkfile_phi = File("2D_gel_D1_t1_n5/polumer_volume_fraction.pvd");
68. vtkfile_stress = File("2D_gel_D1_t1_n5/cauchy_stress.pvd");
69. #initial chemical potential value in the domian

```

```

70. e_u0 = Expression(('0', '0'), degree=1)
71. e_Pi0 = Expression('0', degree=1)
72. e_C0 = Expression('C0', C0 = C0, degree = 1)
73. e_mu0 = Expression('mu0', mu0=mu0, degree=1)
74. assign(w, [interpolate(e_u0, V1), interpolate(e_Pi0, Q1),\
75.             interpolate(e_C0, Q1), interpolate(e_mu0, Q1)])
76. #define govern equation
77. dt = Expression('dt1', degree=0, dt1=0.1)
78. gradv = grad((u-u_n)/dt)
79. W_vis = vis*det(F)*(0.5*inner(gradv*H, (gradv*H+H.T*gradV.T)) -\
80.             1/3* tr(gradv*H)* tr(gradv*H))
81. W_FC= 0.5*Nv*(lambda0**2*Ic-3-2*ln(lambda0**3*det(F)))\
82.       - Cs*(ln(1+1/Cs))-chi/(1+Cs) + Pi*(det(F)-(1+Cs)/J0)
83. Pi0 = -W_FC - W_vis*dt
84. a0 =(derivative(Pi0, w, wt) + mu*C_s)*dx +\
85.      (-Cs*dot(H.T*grad(mu),H.T*grad(mu_)) - Ct*mu_)*dx
86. J1 = derivative(a0, w, dw)
87. # Newton solver parameters
88. parameters={"newton_solver":{"relative_tolerance":1e-4, \
89.                             "absolute_tolerance":1e-6, "maximum_iterations":10}}
90. t = 0.0
91. for n in range(10+89+82):
92.     # Update previous solution
93.     w_n.assign(w)
94.     # Update current time range(0,0.1,1) range(2,1,90) range(90,5,500)
95.     if t < 0.99:
96.         dt.dt1 = 0.1
97.     elif t < 89.99:
98.         dt.dt1 = 1
99.     elif t < 499.9:
100.         dt.dt1 = 5
101.     t += dt.dt1

```

```

102.     if t > 89.9:
103.         mu_D.t = 90.0
104.     else:
105.         mu_D.t = t
106.     print(t)
107.
108.     solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
109.
110.     p1 = plot(w.sub(1), title=str(t))
111.     plt.colorbar(p1)
112.     plt.show()
113.     p1 = plot(w.sub(0), mode = "displacement", title = 'displacement t='+str(t))
114.     plt.colorbar(p1)
115.     plt.show()
116.     vtkfile_u << (w.sub(0), t)
117.     vtkfile_mu << (w.sub(1), t)

```

B.3 针对标度律修正的化学凝胶溶胀模型

```

1.     from __future__ import print_function
2.     import matplotlib.pyplot as plt
3.     from fenics import *
4.     import numpy as np
5.     # Optimization options for the form compiler
6.     parameters["form_compiler"]["cpp_optimize"] = True
7.     parameters["form_compiler"]["representation"] = "uflacs"
8.     parameters["form_compiler"]["quadrature_degree"] = 3
9.     # Create mesh and define function space
10.    nx = 40
11.    ny = 40
12.    mesh = RectangleMesh(Point(0.0,0.0),Point(20.0,20.0),nx,ny)
13.    #define global parameters and initial parameters
14.    C0 = 0.2

```

```

15. lambda0 = (1+C0)**(1/3)
16. Nv = 0.001
17. chi = 0.2
18. mu0 = ln(C0/(1+C0))+1/(1+C0)+chi/(1+C0)**2+Nv*(1/lambda0-\
19.      1/(1+C0)) # mu0 = -0.81943
20. vis = 0.01
21. v_const = 0.588
22. vttotal = -(3*v_const**2-2.5*v_const+0.5)
23. phi0v = (1/(C0+1))*vttotal
24. # Create mesh and define function space
25. V = VectorElement('P',triangle,2)
26. Q = FiniteElement('P',triangle,1)
27. TH = V*Q
28. V1 = FunctionSpace(mesh, V)
29. Q1 = FunctionSpace(mesh, Q)
30. W = FunctionSpace(mesh,TH)
31. # Mark boundary subdomians
32. left = CompiledSubDomain("near(x[0], side) && on_boundary", side = 0.0)
33. right = CompiledSubDomain("near(x[0], side) && on_boundary", side = 20.0)
34. top = CompiledSubDomain("near(x[1], side) && on_boundary", side = 20.0)
35. bottom = CompiledSubDomain("near(x[1], side) && on_boundary", side = 0.0)
36. # Define chemical potential boundary conditions
37. mu_D = Expression("mu00*(1-t/100.0)", degree = 1, mu00 = mu0, t = 0)
38. #fixed left and bottom sides and Define chemical potential boundary conditions
39. bcl = DirichletBC(W.sub(0).sub(0), Constant(0), left)
40. bcb = DirichletBC(W.sub(0).sub(1), Constant(0), bottom)
41. bct_mu = DirichletBC(W.sub(1), mu_D,top)
42. bcr_mu = DirichletBC(W.sub(1), mu_D, right)
43. bcs = [bcl,bcb,bct_mu,bcr_mu]
44. # Define test functions
45. w = Function(W)
46. u, mu = split(w)

```

```

47. wt = TestFunction(W)
48. v, mu_ = split(wt)
49. dw = TrialFunction(W)
50. w_n = Function(W)
51. u_n, mu_n = split(w_n)
52. # Define deformation gradient tensor
53. F = grad(u) + Identity(len(u))
54. I_ = Identity(len(u))
55. Fv = grad(v)
56. F_mu = grad(mu)
57. F_muv = grad(mu_)
58. H = inv(F)
59. C = F.T*F          # Right Cauchy-Green tensor
60. C_2 = C*C
61. # Invariants of deformation tensors
62. Ic = tr(C)
63. Ic2 = 0.5*(tr(C)*tr(C)-tr(C_2))
64. J = det(F)
65. J_v = det(F)*H*Fv
66. C1 = lambda0**3*det(F)-1
67. phi = 1/(1+C1)
68. lambda_ref = (phi**vtotal)/phi0v
69. # Save solution in VTK format
70. vtkfile_u = File("2D_gel_D1_t1_n5/displacement.pvd");
71. vtkfile_mu = File("2D_gel_D1_t1_n5/chemical_potential.pvd");
72. vtkfile_phi = File("2D_gel_D1_t1_n5/polumer_volume_fraction.pvd");
73. vtkfile_stress = File("2D_gel_D1_t1_n5/cauchy_stress.pvd");
74. #initial chemical potential value in the domian
75. e_u0 = Expression(('0', '0'), degree=1)
76. e_mu0 = Expression('mu0', mu0=mu0, degree=1)
77. assign(w, [interpolate(e_u0, V1), interpolate(e_mu0, Q1)])
78. #define govern equation

```



```

79. dt = Expression('dt1', degree=0, dt1=0.1)
80. gradv = grad((u-u_n)/dt)
81. W_vis = vis*det(F)*(0.5*inner(gradv*H, (gradv*H+H.T*gradV.T)) -\
82.         1/3* tr(gradv*H)* tr(gradv*H))
83. W_FC = 0.5*Nv*(lambda0**2*Ic/lambda_ref**2-3-\
84.         2*ln((lambda0/lambda_ref)**3*det(F)))-C1*(ln(1+1/C1))-chi/(1+C1)
85. Pi0 = -W_FC - W_vis*dt
86. a0 = (derivative(Pi0, w, wt) + mu*derivative(C1, w, wt))*dx +\
87.       (-C1*dot(H.T*grad(mu),H.T*grad(mu_)) -\
88.       lambda0**3*det(F)*inner(gradv,H.T)*mu_)*dx
89. J1 = derivative(a0, w, dw)
90. # Newton solver parameters
91. parameters={"newton_solver":{"relative_tolerance":1e-4, \
92.         "absolute_tolerance":1e-6, "maximum_iterations":10}}
93. t = 0.0
94. for n in range(10+89+82):
95.     # Update previous solution
96.     w_n.assign(w)
97.     # Update current time range(0,0.1,1) range(2,1,90) range(90,5,500)
98.     if t < 0.99:
99.         dt.dt1 = 0.1
100.    elif t < 89.99:
101.        dt.dt1 = 1
102.    elif t < 499.9:
103.        dt.dt1 = 5
104.    t += dt.dt1
105.    if t > 89.9:
106.        mu_D.t = 90.0
107.    else:
108.        mu_D.t = t
109.    print(t)
110.

```

```

111.     solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
112.
113.     p1 = plot(w.sub(1),title=str(t))
114.     plt.colorbar(p1)
115.     plt.show()
116.     p1 = plot(w.sub(0), mode = "displacement",title = 'displacement t='+str(t))
117.     plt.colorbar(p1)
118.     plt.show()
119.     vtkfile_u << (w.sub(0), t)
120.     vtkfile_mu << (w.sub(1), t)

```

B.4 针对标度律修正的化学凝胶退溶胀模型

```

1.     from __future__ import print_function
2.     import matplotlib.pyplot as plt
3.     from fenics import *
4.     import numpy as np
5.     # Optimization options for the form compiler
6.     parameters["form_compiler"]["cpp_optimize"] = True
7.     parameters["form_compiler"]["representation"] = "uflacs"
8.     parameters["form_compiler"]["quadrature_degree"] = 3
9.     # Create mesh and define function space
10.    nx = 40
11.    ny = 40
12.    mesh = RectangleMesh(Point(0.0,0.0),Point(20.0,20.0),nx,ny)
13.    #define global parameters and initial parameters
14.    C0 = 0.2
15.    lambda0 = (1+C0)**(1/3)
16.    Nv = 0.001
17.    chi = 0.2
18.    mu0 = ln(C0/(1+C0))+1/(1+C0)+chi/(1+C0)**2+Nv*(1/lambda0-\
19.        1/(1+C0)) # mu0 = -0.81943

```

```

20. vis = 0.01
21. v_const = 0.588
22. vttotal = -(3*v_const**2-2.5*v_const+0.5)
23. phi0v = (1/(C0+1))*vttotal
24. # Create mesh and define function space
25. V = VectorElement('P',triangle,2)
26. Q = FiniteElement('P',triangle,1)
27. TH = V*Q
28. V1 = FunctionSpace(mesh, V)
29. Q1 = FunctionSpace(mesh, Q)
30. W = FunctionSpace(mesh,TH)
31. # Mark boundary subdomians
32. left = CompiledSubDomain("near(x[0], side) && on_boundary", side = 0.0)
33. right = CompiledSubDomain("near(x[0], side) && on_boundary", side = 20.0)
34. top = CompiledSubDomain("near(x[1], side) && on_boundary", side = 20.0)
35. bottom = CompiledSubDomain("near(x[1], side) && on_boundary", side = 0.0)
36. # Define chemical potential boundary conditions
37. mu_D = Expression("mu00*(1+t/100.0)", degree = 1, mu00 = mu0, t = 0)
38. #fixed left and bottom sides and Define chemical potential boundary conditions
39. bcl = DirichletBC(W.sub(0).sub(0), Constant(0), left)
40. bcb = DirichletBC(W.sub(0).sub(1), Constant(0), bottom)
41. bct_mu = DirichletBC(W.sub(1), mu_D,top)
42. bcr_mu = DirichletBC(W.sub(1), mu_D, right)
43. bcs = [bcl,bcb,bct_mu,bcr_mu]
44. # Define test functions
45. w = Function(W)
46. u, mu = split(w)
47. wt = TestFunction(W)
48. v, mu_ = split(wt)
49. dw = TrialFunction(W)
50. w_n = Function(W)
51. u_n, mu_n = split(w_n)

```

```

52.  # Define deformation gradient tensor
53.  F = grad(u) + Identity(len(u))
54.  I_ = Identity(len(u))
55.  Fv = grad(v)
56.  F_mu = grad(mu)
57.  F_muv = grad(mu_)
58.  H = inv(F)
59.  C = F.T*F          # Right Cauchy-Green tensor
60.  C_2 = C*C
61.  # Invariants of deformation tensors
62.  Ic = tr(C)
63.  Ic2 = 0.5*(tr(C)*tr(C)-tr(C_2))
64.  J = det(F)
65.  J_v = det(F)*H*Fv
66.  C1 = lambda0**3*det(F)-1
67.  phi = 1/(1+C1)
68.  lambda_ref = (phi**vtotal)/phi0v
69.  # Save solution in VTK format
70.  vtkfile_u = File("2D_gel_D1_t1_n5/displacement.pvd");
71.  vtkfile_mu = File("2D_gel_D1_t1_n5/chemical_potential.pvd");
72.  vtkfile_phi = File("2D_gel_D1_t1_n5/polumer_volume_fraction.pvd");
73.  vtkfile_stress = File("2D_gel_D1_t1_n5/cauchy_stress.pvd");
74.  #initial chemical potential value in the domian
75.  e_u0 = Expression(('0', '0'), degree=1)
76.  e_mu0 = Expression('mu0', mu0=mu0, degree=1)
77.  assign(w, [interpolate(e_u0, V1), interpolate(e_mu0, Q1)])
78.  #define govern equation
79.  dt = Expression('dt1', degree=0, dt1=0.1)
80.  gradv = grad((u-u_n)/dt)
81.  W_vis = vis*det(F)*(0.5*inner(gradv*H, (gradv*H+H.T*gradV.T)) -\
82.          1/3* tr(gradv*H)* tr(gradv*H))
83.  W_FC = 0.5*Nv*(phi*1.2)**1.07*(Ic/(lambda_ref**2)-3-\

```

```

84.      2*ln(det(F)/( lambda_ref**3))) - C1*(ln(1+1/C1))-chi/(1+C1)
85.  Pi0 = -W_FC - W_vis*dt
86.  a0 = (derivative(Pi0, w, wt) + mu*derivative(C1, w, wt))*dx +\
87.      (-C1*dot(H.T*grad(mu),H.T*grad(mu_)) -\
88.      lambda0**3*det(F)*inner(gradv,H.T)*mu_)*dx
89.  J1 = derivative(a0, w, dw)
90.  # Newton solver parameters
91.  parameters={"newton_solver":{"relative_tolerance":1e-4, \
92.      "absolute_tolerance":1e-6, "maximum_iterations":10}}
93.  t = 0.0
94.  for n in range(10+89+82):
95.      # Update previous solution
96.      w_n.assign(w)
97.      # Update current time range(0,0.1,1) range(2,1,90) range(90,5,500)
98.      if t < 0.99:
99.          dt.dt1 = 0.1
100.     elif t < 89.99:
101.         dt.dt1 = 1
102.     elif t < 499.9:
103.         dt.dt1 = 5
104.     t += dt.dt1
105.     if t > 89.9:
106.         mu_D.t = 90.0
107.     else:
108.         mu_D.t = t
109.     print(t)
110.
111.     solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
112.
113.     p1 = plot(w.sub(1),title=str(t))
114.     plt.colorbar(p1)
115.     plt.show()

```

```

116.    p1 = plot(w.sub(0), mode = "displacement", title = 'disp t='+str(t))
117.    plt.colorbar(p1)
118.    plt.show()
119.    vtkfile_u << (w.sub(0), t)
120.    vtkfile_mu << (w.sub(1), t)

```

B.5 水凝胶薄膜的起皱模型

```

1.    from __future__ import print_function
2.    import matplotlib.pyplot as plt
3.    from fenics import *
4.    import numpy as np
5.    # Optimization options for the form compiler
6.    parameters["form_compiler"]["cpp_optimize"] = True
7.    parameters["form_compiler"]["representation"] = "uflacs"
8.    parameters["form_compiler"]["quadrature_degree"] = 3
9.    # Create mesh and define function space
10.   nx = 100
11.   ny = 52
12.   mesh = RectangleMesh(Point(0.0,0.0),Point(200.0,52.0),nx,ny)
13.   #define global parameters and initial parameters
14.   #Initialize C0 Nv mu0
15.   D = 5 #diffusion coefficient
16.   tol = 1E-14
17.   chi = 0.2
18.   C0_0 = 0.2
19.   C0_1 = 0.2
20.   Nv_0 = 0.001
21.   Nv_1 = 1.0
22.   lambda01 = (1+C0_0)**(1/3)
23.   lambda02 = (1+C0_1)**(1/3)
24.   mu_0 = ln(C0_0/(1+C0_0))+1/(1+C0_0)+chi/(1+C0_0)**2+\
25.       Nv_0*(1/lambda01-1/(1+C0_0))

```

```

26. mu_1 = ln(C0_1/(1+C0_1))+1/(1+C0_1)+chi/(1+C0_1)**2+\
27.     Nv_1*(1/lambda02-1/(1+C0_1))
28. #define different C0 Nv mu0 in defferent domains
29. C0 = Expression('x[1]<=50.0 + tol ? C0_0 : C0_1', degree=0,\
30.     tol = tol, C0_0 = C0_0, C0_1 = C0_1)
31. Nv = Expression('x[1]<=50.0 + tol ? Nv_0 : Nv_1', degree=0,\
32.     tol = tol, Nv_0 = Nv_0, Nv_1 = Nv_1)
33. mu0 = Expression('x[1]<=50.0 + tol ? mu_0 : mu_1', degree=0,\
34.     tol = tol, mu_0 = mu_0, mu_1 = mu_1)
35. lambda0 = Expression('x[1]<=50.0 + tol ? lambda01: lambda02', degree=0,\
36.     tol = tol, lambda01 = lambda01, lambda02 = lambda02)
37. # Create mesh and define function space
38. V = VectorElement('P',triangle,2)
39. Q = FiniteElement('P',triangle,1)
40. TH = V*Q
41. V1 = FunctionSpace(mesh, V)
42. Q1 = FunctionSpace(mesh, Q)
43. W = FunctionSpace(mesh,TH)
44. # Mark boundary subdomians
45. left = CompiledSubDomain("near(x[0], side) && on_boundary", side = 0.0)
46. right = CompiledSubDomain("near(x[0], side) && on_boundary", side = 200.0)
47. top = CompiledSubDomain("near(x[1], side) && on_boundary", side = 52.0)
48. bottom = CompiledSubDomain("near(x[1], side) && on_boundary", side = 0.0)
49. # Define chemical potential boundary conditions
50. mu_D = Expression("mu00*(1+t/100.0)", degree = 1, mu00 = mu0, t = 0)
51. #fixed left and bottom sides and Define chemical potential boundary conditions
52. bcl = DirichletBC(W.sub(0).sub(0), Constant(0), left)
53. bcr = DirichletBC(W.sub(0).sub(0), Constant(0), right)
54. bcb = DirichletBC(W.sub(0).sub(1), Constant(0), bottom)
55. bct_mu = DirichletBC(W.sub(1), mu_D,top)
56. bcs = [bcl,bcr,bcb,bct_mu]
57. # Define test functions

```

```

58. w = Function(W)
59. u, mu = split(w)
60. wt = TestFunction(W)
61. v, mu_ = split(wt)
62. dw = TrialFunction(W)
63. w_n = Function(W)
64. u_n, mu_n = split(w_n)
65. # Define deformation gradient tensor
66. F = grad(u) + Identity(len(u))
67. I_ = Identity(len(u))
68. Fv = grad(v)
69. F_mu = grad(mu)
70. F_muv = grad(mu_)
71. H = inv(F)
72. C = F.T*F          # Right Cauchy-Green tensor
73. C_2 = C*C
74. # Invariants of deformation tensors
75. Ic = tr(C)
76. Ic2 = 0.5*(tr(C)*tr(C)-tr(C_2))
77. J = det(F)
78. J_v = det(F)*H*Fv
79. C1 = lambda0**3*det(F)-1
80. # Save solution in VTK format
81. vtkfile_u = File("2D_gel_D1_wrinkle/displacement.pvd");
82. vtkfile_mu = File("2D_gel_D1_wrinkle/chemical_potential.pvd");
83. vtkfile_phi = File("2D_gel_D1_wrinkle/polimer_volume_fraction.pvd");
84. vtkfile_stress = File("2D_gel_D1_wrinkle/cauchy_stress.pvd");
85. #initial chemical potential value in the domian
86. e_u0 = Expression(('0', '0'), degree=1)
87. #e_mu0 = Expression('mu0', mu0=mu0, degree=1)
88. assign(w, [interpolate(e_u0, V1), interpolate(mu0, Q1)])
89. # Newton solver parameters

```



```

90. parameters={"newton_solver":{"relative_tolerance":1e-4,\
91.         "absolute_tolerance":1e-6, "maximum_iterations":50}}
92. #define govern equation
93. dt = Expression('dt1', degree=0, dt1=0.1)
94. gradv = grad((u-u_n)/dt)
95. W_vis = vis*det(F)*(0.5*inner(gradv*H, (gradv*H+H.T*gradV.T)) -\
96.         1/3* tr(gradv*H)* tr(gradv*H))
97. W_FC = 0.5*Nv*(lambda0**2*Ic-3-2*ln(lambda0**3*det(F))) -\
98.         C1*(ln(1+1/C1))-chi/(1+C1)
99. vis = 0.01
100. Pi0 = -W_FC - W_vis*dt
101. a0 = (derivative(Pi0, w, wt) + mu*derivative(C1, w, wt))*dx + \
102.         (-C1*D*dot(H.T*grad(mu),H.T*grad(mu_)) - \
103.         lambda0**3*det(F)*inner(gradv,H.T)*mu_)*dx
104. J1 = derivative(a0, w, dw)
105. # Newton solver parameters
106. parameters={"newton_solver":{"relative_tolerance":1e-4,
107.         "absolute_tolerance":1e-6, "maximum_iterations":10}}
108. #define the intial and Maximum time increment and Minimum time increment
109. ini_time_int = 0.1
110. max_time_int = 1
111. min_time_int = 0.0001
112. total_time = 200
113. max_steps = 1000
114. #initial time
115. t= ini_time_int
116.
117. for n in range(max_steps):
118.     # Update previous solution
119.     w_n.assign(w)
120.     str1 = "time = " + str(t)
121.     print(str1)

```

```

122.     try:
123.         if t > 89.9:
124.             mu_D.t = 90.0
125.         else:
126.             mu_D.t = t
127.             solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
128.             if dt.dt1 <= 1-1E-6:
129.                 dt.dt1 = dt.dt1*2
130.             else:
131.                 dt.dt1 = max_time_int
132.             str1 = "time increment = " + str(dt.dt1)
133.             print(str1)
134.             t = t + dt.dt1
135.     except RuntimeError:
136.         print("time increment is too large")
137.         t = t - dt.dt1
138.         dt.dt1 = dt.dt1/5
139.         t = t + dt.dt1
140.         if t > 89.9:
141.             mu_D.t = 90.0
142.         else:
143.             mu_D.t = t
144.             solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
145.             if dt.dt1 < 1:
146.                 dt.dt1 = dt.dt1*2
147.
148.             if dt.dt1 < min_time_int:
149.                 break
150.
151.             str1 = "time increment = " + str(dt.dt1)
152.             print(str1)
153.             t = t + dt.dt1

```

```
154.     except RuntimeError:
155.         print("time increment is too large")
156.         t = t - dt.dt1
157.         dt.dt1 = dt.dt1/5
158.         t = t + dt.dt1
159.
160.         if t>89.9:
161.             mu_D.t = 90.0
162.         else:
163.             mu_D.t = t
164.             solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
165.             if dt.dt1 < 1:
166.                 dt.dt1 = dt.dt1*2
167.             if dt.dt1 < min_time_int:
168.                 break
169.             str1 = "time increment = " + str(dt.dt1)
170.             print(str1)
171.             t = t + dt.dt1
172.     except RuntimeError:
173.         print("time increment is too large")
174.         t = t - dt.dt1
175.         dt.dt1 = dt.dt1/5
176.         t = t + dt.dt1
177.         if t>89.9:
178.             mu_D.t = 90.0
179.         else:
180.             mu_D.t = t
181.             solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
182.             if dt.dt1 < 1:
183.                 dt.dt1 = dt.dt1*2
184.             if dt.dt1 < min_time_int:
185.                 break
```

```

186.     str1 = "time increment = " + str(dt.dt1)
187.     print(str1)
188.     t = t + dt.dt1
189.     #postprocessing
190.     p1 = plot(w.sub(1), title='chemical potential t='+str(t))
191.     plt.colorbar(p1)
192.     plt.show()
193.     p1 = plot(w.sub(0), mode = "displacement",title = 'displacement t='+str(t))
194.     plt.colorbar(p1)
195.     plt.show()
196.     vtkfile_u << (w.sub(0), t)
197.     vtkfile_mu << (w.sub(1), t)
198.     if t>=total_time:
199.         break

```

B.6 化学凝胶长条溶胀模型

```

1.     import matplotlib.pyplot as plt
2.     from fenics import *
3.     #from mshr import *
4.     from math import log
5.     import numpy as np
6.     # Optimization options for the form compiler
7.     parameters["form_compiler"]["cpp_optimize"] = True
8.     parameters["form_compiler"]["representation"] = "uflacs"
9.     parameters["form_compiler"]["quadrature_degree"] = 3
10.    # Create mesh and define function space
11.    mesh = Mesh("two_solvents0415.xml")
12.    #define global parameters and initial parameters
13.    C10 = 0.2
14.    C20 = 0.001
15.    C1_final = 0.001
16.    C2_final = 1.0

```

```

17. J0 = 1+C10+C20
18. lambda0 = (J0)**(1/3)
19. J_final = 1+C1_final+C2_final
20. lambda_final = J_final**(1/3)
21. Nv = 0.01
22. chi1p = 0
23. chi2p = 0
24. chi12 = 0
25. mu10 = Nv*(1/lambda0-1/J0)+ln(C10/J0)+(1+chi1p)/J0-\
26.         (chi1p*C10+chi2p*C20)/J0**2+chi12*C20/J0-\
27.         chi12*C10*C20/J0**2
28. mu20 = Nv*(1/lambda0-1/J0)+ln(C20/J0)+(1+chi2p)/J0-\
29.         (chi1p*C10+chi2p*C20)/J0**2+chi12*C10/J0-\
30.         chi12*C10*C20/J0**2
31. mu1_final = Nv*(1/lambda_final-1/J_final)+ln(C1_final/J_final)+\
32.         (1+chi1p)/J_final-(chi1p*C1_final+chi2p*C2_final)/J_final**2+\
33.         chi12*C2_final/J_final-chi12*C1_final*C2_final/J_final**2
34. mu2_final = Nv*(1/lambda_final-1/J_final)+ln(C1_final/J_final)+\
35.         (1+chi1p)/J_final-(chi1p*C1_final+chi2p*C2_final)/J_final**2+\
36.         chi12*C2_final/J_final-chi12*C1_final*C2_final/J_final**2
37. vis = 0.01
38. D1 = 0.005 #diffusion constant
39. D2 = 0.1
40. # Create mesh and define function space
41. plot(mesh,title='mesh')
42. plt.show()
43. # Create mesh and define function space
44. V = VectorElement('P',tetrahedron,1)
45. Q = FiniteElement('P',tetrahedron,1)
46. TH = MixedElement([V, Q, Q, Q, Q, Q])
47. V1 = FunctionSpace(mesh, V)
48. Q1 = FunctionSpace(mesh, Q)

```

```

49. W = FunctionSpace(mesh,TH)
50. # unknown fields: u, Pi, Cs1, mu1, Cs2, mu2
51. # Mark boundary subdomians
52. left = CompiledSubDomain("near(x[0], side) && on_boundary", side = 0.0)
53. right = CompiledSubDomain("near(x[0], side) && on_boundary", side = 50.0)
54. up1 = CompiledSubDomain(" x[0]>1E-10 && x[1]>1E-10 && x[2]>1E-10 && \
55.     x[0]< 50-(1E-10) && on_boundary", side = 0.0)
56. up2 = CompiledSubDomain("near(x[1], side1) && near(x[2], side2) && \
57.     on_boundary", side1 = 0.0, side2 = 4.0)
58. bottom = CompiledSubDomain("near(x[2], side) && on_boundary", side = 0.0)
59. front = CompiledSubDomain("near(x[1], side) && on_boundary", side = 0.0)
60. left_side = CompiledSubDomain("x[0]<=side1 && near(x[2], side2)
61.     && on_boundary", side = 0.0)
62. # Define chemical potential boundary conditions
63. C1_bc_RU = Expression('C10+(C1_final-C10)*t/100', degree=1,
64.     C10 = C10, C1_final = C1_final, t=0)
65. C2_bc_RU = Expression('C20+(C2_final-C20)*t/100', degree=1,
66.     C20 = C20, C2_final = C2_final, t=0)
67. J_bc = Expression('1+C1_bc_RU+C2_bc_RU', degree=1,
68.     C1_bc_RU = C1_bc_RU, C2_bc_RU = C2_bc_RU)
69. mu1_bc_RU = Expression('Nv*(1/pow(J_bc,1/3)-1/J_bc) + \
70.     log10(C1_bc/J_bc)/log10(2.71828) + 1/J_bc', degree=1, Nv = Nv, \
71.     J_bc = J_bc, C1_bc = C1_bc_RU, C2_bc = C2_bc_RU, chi1p = chi1p)
72. mu2_bc_RU = Expression('Nv*(1/pow(J_bc,1/3)-1/J_bc) + \
73.     log10(C2_bc/J_bc)/log10(2.71828) + 1/J_bc', degree=1, Nv = Nv, \
74.     J_bc = J_bc, C1_bc = C1_bc_RU, C2_bc = C2_bc_RU, chi2p = chi2p)
75. # Define test functions
76. w = Function(W)
77. u, Pi, Cs1, mu1, Cs2, mu2 = split(w)
78. wt = TestFunction(W)
79. v, Pi_, Cs1_, mu1_, Cs2_, mu2_ = split(wt)
80. dw = TrialFunction(W)

```

```

81. w_n = Function(W)
82. u_n, Pi_n, Cs1_n, mu1_n, Cs2_n, mu2_n = split(w_n)
83. #fixed left and bottom sides and Define chemical potential boundary conditions
84. bcl1 = DirichletBC(W.sub(0).sub(0), Constant(0), left)
85. bcl2 = DirichletBC(W.sub(0).sub(2), Constant(0), left)
86. bcf = DirichletBC(W.sub(0).sub(1), Constant(0), front)
87. bcu_mu1 = DirichletBC(W.sub(3), mu1_D, up1)
88. bcu_mu2 = DirichletBC(W.sub(5), mu2_D, up1)
89. bcb_mu1 = DirichletBC(W.sub(3), mu1_D, bottom)
90. bcb_mu2 = DirichletBC(W.sub(5), mu2_D, bottom)
91. bcs = [bcl1,bcl2,bcf, bcu_mu1,bcu_mu2,bcb_mu1,bcb_mu2]
92. # Define deformation gradient tensor
93. F = grad(u) + Identity(len(u))
94. I_ = Identity(len(u))
95. Fv = grad(v)
96. H = inv(F)
97. C = F.T*F          # Right Cauchy-Green tensor
98. C_2 = C*C
99. # Invariants of deformation tensors
100. Ic = tr(C)
101. Ic2 = 0.5*(tr(C)*tr(C)-tr(C_2))
102. J = det(F)
103. J_v = det(F)*H*Fv
104. # Save solution in VTK format
105. vtkfile_u = File("3D_gel_D0_1/displacement.pvd");
106. vtkfile_Cs1 = File("3D_gel_D0_1/concentration1.pvd");
107. vtkfile_mu1 = File("3D_gel_D0_1/chemical_potential1.pvd");
108. vtkfile_Cs2 = File("3D_gel_D0_1/concentration2.pvd");
109. vtkfile_mu2 = File("3D_gel_D0_1/chemical_potential2.pvd");
110. vtkfile_stress = File("3D_gel_D0_1/cauchy_stress.pvd");
111. #initial chemical potential value in the domian
112. e_u0 = Expression(('0', '0','0'), degree=1)

```

```

113. e_Pi0 = Expression('0', degree=1)
114. e_C1 = Expression('C0', C0 = C10, degree = 1)
115. e_mu1 = Expression('mu0', mu0=mu10, degree=1)
116. e_C2 = Expression('C0', C0 = C20, degree = 1)
117. e_mu2 = Expression('mu0', mu0=mu20, degree=1)
118. assign(w, [interpolate(e_u0, V1), interpolate(e_Pi0, Q1), interpolate(e_C1, Q1),\
119.             interpolate(e_mu1, Q1), interpolate(e_C2, Q1), interpolate(e_mu2, Q1)])
120. cs1 = Cs1/(det(F)*lambda0**3)
121. cs2 = Cs2/(det(F)*lambda0**3)
122. dt = Expression('dt1', degree=0, dt1=0.1)
123. Cs1t = (Cs1-Cs1_n)/dt
124. Cs2t = (Cs2-Cs2_n)/dt
125. W_FC = 0.5*Nv*(lambda0**2*Ic-3-2*ln((lambda0)**3*det(F))) +
126.         Cs1*ln(cs1)+Cs2*ln(cs2)+chi1p*cs1+chi2p*cs2+chi12*Cs1*cs2
127. Pi0 = -W_FC
128. IMCOMPRES = Pi*(det(F)-(1+Cs1+Cs2)/J0)
129. a0 = (derivative(Pi0, w, wt) + mu1*Cs1_ + mu2*Cs2_)/J0*dx
130.     + derivative(IMCOMPRES, w, wt)*dx+\
131.     + (-1*D1*Cs1/J0*dot(H.T*grad(mu1),H.T*grad(mu1_)) - C1t/J0*mu1_)*dx \
132.     + (-1*D2*Cs2/J0*dot(H.T*grad(mu2),H.T*grad(mu2_)) - C2t/J0*mu2_)*dx
133. J1 = derivative(a0, w, dw)
134. # Newton solver parameters
135. parameters={"newton_solver":{"relative_tolerance":1e-4,\
136.                               "absolute_tolerance":1e-6, "maximum_iterations":20}}
137. #define the intial and Maximum time increment and allowed Minimum time increment
138. ini_time_int = 0.001
139. max_time_int = 1
140. min_time_int = 0.0001
141. total_time = 200
142. max_steps = 1000
143. #initial time
144. t= ini_time_int

```



```
145. for n in range(max_steps):
146.     # Update previous solution
147.     w_n.assign(w)
148.     str1 = "time = " + str(t)
149.     print(str1)
150.     try:
151.         if t >= 96.0:
152.             C1_bc_RU.t = 96
153.             C2_bc_RU.t = 96
154.         else:
155.             C1_bc_RU.t = t
156.             C2_bc_RU.t = t
157.         solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
158.         if dt.dt1 <= 1-1E-6:
159.             dt.dt1 = dt.dt1*2
160.         else:
161.             dt.dt1 = max_time_int
162.         str1 = "time increment = " + str(dt.dt1)
163.         print(str1)
164.         t = t + dt.dt1
165.     except RuntimeError:
166.         print("time increment is too large")
167.         t = t - dt.dt1
168.         dt.dt1 = dt.dt1/5
169.         t = t + dt.dt1
170.         if t >= 96.0:
171.             C1_bc_RU.t = 96
172.             C2_bc_RU.t = 96
173.         else:
174.             C1_bc_RU.t = t
175.             C2_bc_RU.t = t
176.         solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
```

```

177.     if dt.dt1 < 1:
178.         dt.dt1 = dt.dt1*2
179.     if dt.dt1 < min_time_int:
180.         break
181.     str1 = "time increment = " + str(dt.dt1)
182.     print(str1)
183.     t = t + dt.dt1
184. except RuntimeError:
185.     print("time increment is too large")
186.     t = t - dt.dt1
187.     dt.dt1 = dt.dt1/5
188.     t = t + dt.dt1
189.     if t >= 96.0:
190.         C1_bc_RU.t = 96
191.         C2_bc_RU.t = 96
192.     else:
193.         C1_bc_RU.t = t
194.         C2_bc_RU.t = t
195.     solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
196.     if dt.dt1 < 1:
197.         dt.dt1 = dt.dt1*2
198.     if dt.dt1 < min_time_int:
199.         break
200.     str1 = "time increment = " + str(dt.dt1)
201.     print(str1)
202.     t = t + dt.dt1
203. except RuntimeError:
204.     print("time increment is too large")
205.     t = t - dt.dt1
206.     dt.dt1 = dt.dt1/5
207.     t = t + dt.dt1
208.     if t >= 96.0:

```

```

209.         C1_bc_RU.t = 96
210.         C2_bc_RU.t = 96
211.     else:
212.         C1_bc_RU.t = t
213.         C2_bc_RU.t = t
214.     solve(a0==0, w, bcs, J = J1, solver_parameters=parameters)
215.     if dt.dt1 < 1:
216.         dt.dt1 = dt.dt1*2
217.     if dt.dt1 < min_time_int:
218.         break
219.     str1 = "time increment = " + str(dt.dt1)
220.     print(str1)
221.     t = t + dt.dt1
222.
223.     p1 = plot(w.sub(3),title=str(t))
224.     plt.colorbar(p1)
225.     plt.show()
226.     p1 = plot(w.sub(0), mode = "displacement",title = 'disp t='+str(t))
227.     plt.colorbar(plot((w.sub(0).sub(0)**2+w.sub(0).sub(1)**2+w.sub(0).sub(2)**2)
228.     **0.5))
228.     plt.show()
229.     vtkfile_u << (w.sub(0), t)
230.     vtkfile_C1 << (w.sub(2), t)
231.     vtkfile_mu1 << (w.sub(3), t)
232.     vtkfile_C2 << (w.sub(4), t)
233.     vtkfile_mu2 << (w.sub(5), t)
234.     if t>=total_time:
235.         break

```

攻读硕士学位期间发表的论文及其它成果

（一）参与的科研项目及获奖情况

- [1] 章肖阳，徐翌，王志备. 混凝土夹心拱的综合力学问题有限元分析. 广东省第一届有限元建模大赛三等奖, 2018.
- [2] 章肖阳，刘伟轩，何闯. 大型户外单立柱广告牌在 15 级风载荷作用下失效问题的静力学、流固耦合及冲击动力学有限元分析. 广东省第二届有限元建模大赛二等奖, 2019.
- [3] 章肖阳，肖思[#]，洪伟[#]. 基于深度学习的裂纹尖端复杂应力场预测. 拟发表期刊论文.

哈尔滨工业大学与南方科技大学联合培养研究生学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于开源有限元的水凝胶力学行为研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：章肖阳 日期：2020 年 6 月 6 日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

(1) 学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；(2) 学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；(3) 研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为南方科技大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：章肖阳 日期：2020 年 6 月 6 日

导师签名：洪伟 日期：2020 年 6 月 6 日

致 谢

仲春时节，风和景明，芳草鲜美，落英缤纷，历时期年，论文初成，童稚至今，寒窗二十余载，及时疫之年，思忖良多，于鹏城作此篇，铭谢师友，亦寄吾志。

受教于南科已近二载，于航天之学院，习力学之理论，眼界学识，均有增益。南科者，立时代前沿之地，应学府革新之潮，收世界科技之利器，纳天下博学之英才。使集群电镜，多于湖广之山川；院士千人，多于苏浙之江河；莘莘学子，多于在庾之粟粒；科研硕果，多于周身之帛缕。

恩师洪伟，高山仰止，熠熠其华，灼灼其理，学而不厌，诲人不倦，师外长技，学成报国，治学为人，言传身教。

导师肖思，求学于清华，施教于南科，学习工作，科研论文，悉以咨之，事无巨细，皆细查详答，授业解惑，关怀备至。

同窗徐墨、屈冲、海涛、一鸣，皆为俊才，性行淑均，品行纯良，同行二载，互帮互衬，愿共步光明之道，具遂凌之云志。

明明如月，何时可掇？前路漫漫，其途渺渺，不失初心，惟愿求索。披坚执锐，乘风破浪，心之所向，无问沧桑。