基于优化消去的大规模 RC 网络模型降阶算法研究

Model Order Reduction for Large Scale RC Networks Based on Optimal Elimination

(申请清华大学工学硕士学位论文)

培 养 单 位 ： 计算机科学与技术系

学　　科 ： 计算机科学与技术

研 究 生 ： 程 康

指 导 教 师 ： 喻 文 健 副 教 授

二〇一二年五月

基于优化消去的大规模RC网络模型降阶算法研究

程　康

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，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；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后遵守此规定）

作者签名：_____ 导师签名：_____

日　　期：_____ 日　　期：_____

摘 要

随着集成电路的不断发展，制造工艺日新月异，特征尺寸迈入纳米级，晶体管已经数十亿计，工作频率也在不断提升，互连线对电路性能的影响已经超过了晶体管，成为了主要因素。能够准确高效处理互连线的软件成为了 EDA 的重要部分，模型降阶是解决 EDA 中大规模 RC 网络问题的关键所在。模型降阶技术已广泛应用于集成电路的建模、仿真和分析的方方面面，但对电路后仿分析中遇到的大规模、多端口 RC 网络的降阶仍是一个难以解决的问题。

由于问题规模的不断增大，在 RC 网络处理过程中模型降阶算法的作用越来越突出，并且起到了不错的效果。但是，近年来由于问题规模的进一步增大和集成电路构造越来越复杂，对模型降阶算法在计算效率和应对问题特殊要求的能力提出了更高的要求，导致已有的一些基于投影法和平衡截断法的模型降阶算法不再适用。近年来，基于节点消去的 RC 网络模型降阶方法受到重视，并取得了一定的效果。

本文指出，由于稀疏性受到破坏，网络的元件数（或对应方程的矩阵非零元）随着节点消去的过程有时不但不减小，反而会显著增加。因此在消去顺序确定的情况下，可能存在一个最优的剩余电路节点数，使得降阶后的网络求解速度达到最优。基于这个最优化消去的思想，本文首先采用统计建模的方法得到 RC 网络求解时间与节点数、非零元数的近似关系，然后在对实际电路进行模型降阶时用节点消去法快速得到剩余节点数与非零元的关系，从中选出最优的剩余网络节点数并根据它实施模型降阶。实验结果表明，采用这种最优节点消去策略得到的降阶网络不造成精度损失，且使电路求解速度比已有方法快 2~5 倍。

关键词：电路后仿、模型降阶、节点消去、数据拟合

Abstract

Along with the development of the integrated circuit, manufacturing process changes with each new day, the characteristic dimensions entering to the nanometer level, the transistor has billions, working frequency also is rising, to interconnect influence on the performance of the circuit is more than a transistor, become a major factor. Can accurate efficient processing interconnect software became an important part of the EDA, model order reduction (MOR) is to solve the EDA medium to large RC network of key issues. MOR has been widely used in modeling and simulation of integrated circuits. MOR for large scale RC networks resulting from post-layout circuit analysis is still a difficult problem.

Because of the size of the problem is continuously increasing, and in the process of RC network model of the reduced order algorithm functions are becoming more and more outstanding, and played a pretty good effect, but in recent years because of the size of the problem further increase and integrated circuit structure is more and more complicated, the reduced order model algorithm in calculation and approach to problems of the special requirements for the ability to put forward higher request, has led to some of the projection method based on the model of the law and balance stage order reduction algorithm is no longer apply. Elimination based MOR methods has attracted focus.

However, in practice, the number of elements in a network may not necessarily reduces during the elimination due to the degradation of sparsity. Given elimination order, in theory there is an optimal elimination, with which the resulting network has minimal elements. In this paper, we propose to find this optimal elimination based on symbolic analysis and statistical analysis.

Experiments show that with such approach the solution time of reduced network can be reduced by a factor of 2~5.
Keywords: Post Simulation，Model Order Reduction，Node Elimination，Data Fitting

目 录

第一章 前言

集成电路支撑着整个信息产业，电子设计自动化（Electronic Design Automation, EDA）更是集成电路高速发展和不断进步的支柱与有力保障。同时集成电路产业作为信息产业的基石，已经对人类日常生活和社会的整体进步产生了不可估量并且无比深远的影响。现代金融、军事、通信、数字娱乐、计算机与互联网等方面都需要集成电路的支撑。集成电路的制造工艺日新月异，特征尺寸迈入纳米级，晶体管已经数十亿计，工作频率也在不断提升，互连线对电路性能的影响已经超过了晶体管，成为了主要因素。能够准确高效处理互连线的软件成为了 EDA 的重要部分，模型降阶是解决 EDA 中大规模 RC 网络问题的关键所在[2]。本章将从回顾集成电路的历史和现状开始，简要描述模型降阶方法在集成电路互连线处理中的应用，最后介绍本文的主要贡献和论文的组织结构。

1.1 集成电路与模型降阶的发展

1.1.1 集成电路发展的历史及现状

自从美国德州仪器公司于 1958 年制造了全球第一块集成电路后，人类开始迈进了信息化时代。2005 年世界集成电路市场规模为 2357 亿美元，并且直至 2010 年保持了年均超过 10%的高增长，总规模达到了 4247 亿美元，并且贡献了国民生产总值增长的 65%以上。从第一块集成电路诞生到现阶段，集成电路从小规模已经发展到超大规模（VLSI）和特大规模（ULSI）阶段。根据 2011 年半导体协会发布的国际半导体技术蓝图报告（International Technology Roadmap for Semiconductors, ITRS 2012）[1]预测集成电路仍将保持特征尺寸（1/2 pitch）每年缩小 10％的高速度发展。ITRS 2011 中给出的部分预测数据如图 1 所示。

Year of Production	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026
Flash ½ Pitch (nm) (un-contacted Poly)([2]	22	20	18	17	15	14.2	13.0	11.9	10.9	10.0	8.9	8.0	8.0	8.0	8.0	8.0
DRAM ½ Pitch (nm) (contacted)[1,2]	36	32	28	25	23	20.0	17.9	15.9	14.2	12.6	11.3	10.0	8.9	8.0	7.1	6.3
MPU/ASIC Metal 1 (M1) ½ Pitch (nm)[1,2]	38	32	27	24	21	18.9	16.9	15.0	13.4	11.9	10.6	9.5	8.4	7.5	6.7	6.0
MPU High-Performance Printed Gate Length (GLpr) (nm) ††[1]	35	31	28	25	22	19.8	17.7	15.7	14.0	12.5	11.1	9.9	8.8	7.9	6.79	5.87
MPU High-Performance Physical Gate Length (GLph) (nm)[1]	24	22	20	18	17	15.3	14.0	12.8	11.7	10.6	9.7	8.9	8.1	7.4	6.6	5.9
Logic (Low-volume Microprocessor) High-performance ‡																
Generation at Introduction	p13h	p13h	p16h	p16h	p16h	p19h	p19h	p19h	p22h	p22h	p22h	p25h	p25h	p25h	p28h	p28h
Functions per chip at introduction (million transistors)	8,848	8,848	17,696	17,696	17,696	35,391	35,391	35,391	70,782	70,782	70,782	141,564	141,564	141,564	283,128	283,128
Chip size at introduction (mm^2)	520	368	520	413	328	520	413	328	520	413	328	520	413	328	520	413
Generation at production **	p11h	p11h	p13h	p13h	p13h	p16h	p16h	p16h	p19h	p19h	p19h	p22h	p22h	p22h	p25h	p25h
Functions per chip at production (million transistors)	4,424	4,424	8,848	8,848	8,848	17,696	17,696	17,696	35,391	35,391	35,391	70,782	70,782	70,782	141,564	141,564
Chip size at production (mm^2) §§	260	184	260	206	164	260	206	164	260	206	164	260	206	164	260	206
OH % of Total Chip Area	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%	29.5%
Logic Core+SRAM (Without OH Average Density (Ms/cm2)	2,414	3,414	4,828	6,083	7,664	9,656	12,166	15,328	19,312	24,332	30,656	38,625	48,664	61,313	77,249	97,328
High-performance MPU Mtransistors/cm^2 at introduction and production (including on-chip SRAM) ‡	1,701	2,406	3,403	4,287	5,402	6,806	8,575	10,804	13,612	17,150	21,608	27,224	34,300	43,215	54,448	68,600
ASIC																
ASIC usable Mtransistors/cm^2 (auto layout)	1,701	2,406	3,403	4,287	5,402	6,806	8,575	10,804	13,612	17,150	21,608	27,224	34,300	43,215	54,448	68,600
ASIC max chip size at production (mm^2) (maximum lithographic field size)	858	858	858	858	858	858	858	858	858	858	858	858	858	858	858	858
ASIC maximum functions per chip at production (Mtransistors/chip) (fit in maximum lithographic field size)	14,599	20,646	29,198	36,787	46,348	58,395	73,573	92,697	116,790	147,147	185,393	233,581	294,293	370,786	467,162	588,587

图 1.1　2012 年国际半导体技术蓝图报告 ITRS 中的预测数据[1]

集成电路设计的流程如图 1.2 所示，大致可以分为前端设计及后端设计两大部分，前者通过功能描述得到寄存器/传输级描述，通过逻辑综合得到门级网表，后者根据网表通过布局规划、布局、布线得到最后的物理版图。但是因为收到互连线和器件的电磁寄生效应的影响，这个过程很难一次完成，得到的结果往往达不到预期要求，需要多次迭代才能完成最终的设计。得到物理版图后，要进行寄生参数提取，利用得到的电容、电阻、电感等信息进行电路仿真，根据仿真结果对网表进行调整，重新布局、布线，再生成版图。

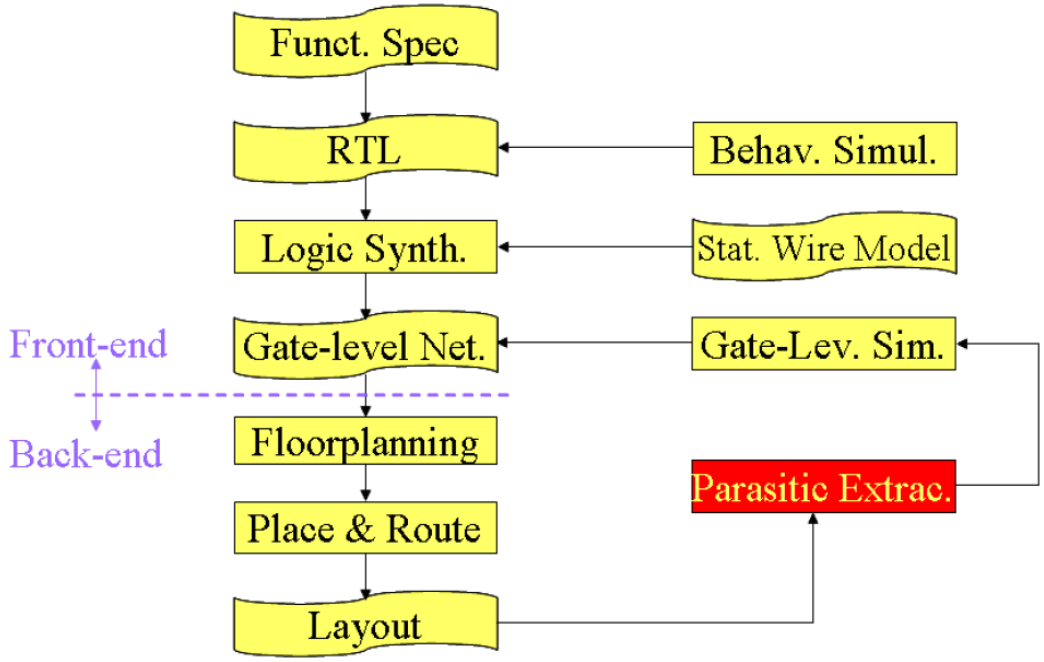


图 1.2 典型集成电路设计流程

1.1.2模型降阶在集成电路设计中的应用

由于特征尺寸的不断减小和晶体管数量的极大增长，在物理设计和之后的“后仿”中经常需要进行时序分析等涉及大规模 RC 网络的工作，而模型降阶方法是处理大规模 RC 网络的有效方法之一。互连线通过标准的节点分析后可以通过一阶或者二阶的微分方程组来表示[7][8-10]，其中的状态变量数量可能达到数十万乃至更高。进行模型降阶就是将大规模的系统用一个规模较小的降阶系统来替代，以达到提高效率、增强可处理性的目的。进行模型降阶时不光要从数学上，而且要从物理上兼顾原系统的特性。也就是说不止要保持原系统的输入与输出之间的关系，更要能够保持原系统的无源性[11]和特殊的系统结构[5][6]。由于模型降阶是一个数值计算的过程，所以计算中的数值稳定性是必须考虑的重要问题。稳定的降阶方法能够将误差控制在一个能够接受的范围内，最大的保证计算精度，从保证降阶系统对原系统的逼近程度，反之，不稳定的降阶方法则会产生各种各样的问题[14][47][48]。

模型降阶方法主要需要考虑以下几个方面的问题：保证输入输出关系的逼近精度、保持原系统的无源性、保持原系统特殊的系统结构以及数值稳定性。应用于 RC 网络中的模型降阶方法可以分为两类，一是平衡截断法 [35][36]，二是矩匹配法。对于线性系统来说，外部输入影响内部状态，同时内部状态也会对系统的输出产生影响。外部输入对于内部状态影响的大小称为可控性，内部状态对于系统输出的影响称为可观性。平衡截断法的本质就是将对原系统输入与输出关系影响比较小的内部状态舍去，从而降低系统的规模。但是内部状态并不一定是同时具有较小的可控性与可观性，有时可控性强可观性弱，有时反之，这样就很难取舍。平衡截断法中的平衡指的是通过对原系统的线性变换，求得其可控性与可观性的格拉姆矩阵，使得系统内部状态的可控性与可观性格拉姆矩阵相等，这样就可以将可控性与可观性都不强的内部状态舍去，从而达到降阶的目的。平衡截断法的优势在于有理论的误差界，但是其计算复杂度很高，并不适用于规模特别大的系统。

针对平衡截断法的上述特点，有人提出了基于迭代的快速平衡阶段方法。通过将 Krylov 子空间方法与平衡阶段法相结合来近似的两者的格拉姆矩阵，或者用数值积分求解格拉姆矩阵[37-46]。这些快速方法虽然降低了计算复杂度，

由于使用的是近似方法，所以误差界无法保证。虽然平衡截断类方法取得了一定的进展，但是由于其理解和实现都比较复杂，所以应用范围有限。

矩匹配类算法是在处理大规模 RC 网络方面应用最广泛的算法[12][13][14-23][26-34]。矩匹配就是以低阶的有理多项式函数来逼近高阶函数，并保证两者传递函数泰勒展开一定的匹配精度，矩指的是传递函数的泰勒展开系数。目前应用比较广泛的是基于 Krylov 子空间投影的矩匹配模型降阶方法[14-23][26-34]。Krylov 子空间投影法能够通过投影保证矩匹配，所以被叫做“隐式”方法。PRIMA 方法[28]也是 Krylov 子空间降阶方法，它借助合同变换来保持原系统的无源性[24][25]。相对于 PRIMA 方法，SPRIM 方法[30]能够保持原系统的块结构。

1.2 集成电路发展、互连电路分析仿真及模型降阶面临的挑战

1.2.1 集成电路发展面临的挑战

集成电路发展到现阶段，面临的主要问题有以下几点[1]:

（1）集成电路的规模和复杂度不断增大。按照摩尔定理，芯片集成度（芯片上可容纳的晶体管数目）可以认为是以每 18 个月增加一倍的速度高速增长，2011 年，商业中央处理器（Central Processing Unit，CPU）包含超过二十亿的晶体管；图形处理器（Graphic Processing Unit，GPU）则超过了三十亿个晶体管[2]。不同系统的集成也使得集成电路的复杂度不断增加，如 System-on-a-Chip（SoC），Network-on-a-Chip（NoC），三维封装和三维集成电路（Three-Dimensional ICs）将不同类型电路（如数字电路、射频电路、模拟电路、存储器等）整合在一起。

（2）集成电路的工作频率不断提高。虽然目前 ASIC、CPU 等高性能数字电路并没有向高频率方向发展，而是偏向于多核、众核、三维芯片等方向发展，微处理器的时钟频率也已经由由 80 年代的数 MHz 增长到了目前的数 GHz。

（3）集成电路器件的特征尺寸在不断减小。进入 21 世纪，集成电路的特征尺寸已经从 0.18、0.13 微米、90nm、 65nm，发展到 45nm 以下，这给制造工艺提出了更高的要求。

（4）功耗、性能和价格之间的矛盾的日渐突出。随着芯片尺寸逐渐减小，要设计出同低功耗、高性能和低价格的集成电路产品已经变得越来越困难。

（5）测试成本不断提高，可制造性设计问题也越来越严重。芯片的测试成本在芯片开发总成本中占了比较大的比例，芯片规模的不断增大，使降低测试成本成为集成电路开发中的一大挑战。同时，随着芯片特征尺寸的减小，芯片的可制造性设计问题也越来越严重。

1.2.2 互连电路分析仿真面临的挑战

（1）互连电路现在已经非常庞大，而今后还将不断的增长。随着工艺尺寸的减小和器件数目的增加，互连电路的规模已经达到了非常大的程度。同时，互连电路之间的电磁耦合数量更是达到了惊人的地步，以至于等效出的电路通常能达到百万阶以上[4][5]。传统的电路分析及模拟工具无法对其进行有效的处理，而对互连电路的处理是寄生参数提取和电路仿真中非常重要和关键的环节[3-5]。

（2）互连线的电磁耦合现象日趋严重。在集成电路还在小规模、低集成度的初级阶段，电信号可以认为是通过连接器件的互连线理想传输的，但是随着半导体特征尺寸的减小，互连线之间的电磁耦合现象越来越严重，电信号已经不能认为是通过互连线理想传输，其在传输过程中会产生延迟、衰减和震荡等现象，其中互连线的延迟已经超过了门延迟，成为决定电路延时的主要因素，严重的影响整个电路的性能[4]。

1.2.3 模型降阶方法面临的挑战

（1）模型降阶面对的问题规模在不断增长。由于互连电路的复杂度和电磁耦合现象的增长，使得需要处理的 RC 网络非常庞大，一些现有的模型降阶方法无法有效的处理这么大规模的问题。

（2）集成电路结构和功能的复杂化对模型降阶提出了新的要求。随着集成电路多核、多功能集成成为发展的趋势，需要模型降阶方法能够更好的应对例如保持无源性、保证特殊结构等要求。

1.3 论文的主要贡献和组织结构

随着集成电路规模的不断扩大和特征尺寸的不断缩小，导致用于互连分析和电路仿真的 RC 网络规模越来越大，不仅直接处理非常困难，很多模型降阶技术也不再适用于规模如此大的问题。同时由于集成电路的结构也越来越复杂，

同样对模型降阶技术提出了新的要求。本文对用于大规模稀疏 RC 网络的模型降阶问题进行了相关研究工作，主要研究内容及成果如下：

1.采用合适的稀疏矩阵排序技术，减少矩阵处理过程中的填入元。稀疏矩阵在处理的过程中，不可避免的会出现不同程度的 fill-in 现象，对于我们所处理的大规模稀疏 RC 网络，其稀疏性是我们希望能够保留的很好的性质，但是在处理过程中往往会破坏原始系统的稀疏性，产生非常稠密甚至是一个满阵，这样会大大的影响算法的效率，同时对存储空间的需求也很难满足。采用合适的稀疏矩阵排序技术，能够尽量减少填入现象对算法的影响，提高算法的效率。考虑到电路后仿的实际情况，经过实验对比，发现使用 CAMD 方法[54]进行排序效果较好。

2.采用最优消去的策略，寻找降阶过程中效果最好的消去程度。采用节点消去法或者投影法对 RC 网络模型进行消去的过程中，RC 矩阵的非零元与维度之间是一个不规则变化的关系，根据处理对象的不同，差异可能会很大，有时因为填入现象的严重，使得维度降低后非零元数量上升很多。举例来说，对于一个维度降为原始系统十分之一，但是非零元数量却增长到原始系统 3 倍的系统，这样的降阶是否能够带来运算速度上的提升并不能确定。本文采取统计建模的方法，通过大规模的实验，拟合得到降阶系统的维度、非零元与处理时间的关系，利用这个关系得到降阶系统的最佳降阶程度。通过实验证明，采用此策略，能够使得到的降阶系统比用现有降阶方法得到的降阶系统，在求解时间上有 2 至 5 倍的加速。

本文首先回顾了集成电路发展的历史、现状及面临的挑战，简要介绍了集成电路的设计流程和模型降阶方法在互连电路分析仿真中的应用。第二章对现有用于互连电路处理的模型降阶方法进行了归纳和分析。第三章提出了基于最优消去的大规模 RC 网络模型降阶方法。最后是总结和展望。

第二章 应用于集成电路设计的主要模型降阶方法

2.1 经典的模型降阶方法

2.1.1 基础的Krylov投影降阶方法

在这一小节，主要介绍最基础的 Krylov 投影降阶方法。

（1）Arnoldi 算法

在此给出一个常见的 n 阶单输入输出系统

$$\begin{cases} \widehat{\boldsymbol{E}}\dfrac{d\boldsymbol{x}(t)}{dt} = \widehat{\boldsymbol{A}}\boldsymbol{x}(t) + \widehat{\boldsymbol{b}}u(t) \\ y(t) = \hat{\boldsymbol{c}}^T\boldsymbol{x}(t) \end{cases}$$

其中$\widehat{\boldsymbol{E}}$，$\widehat{\boldsymbol{A}} \in \boldsymbol{R}^{n\times n}$，$\widehat{\boldsymbol{b}}$，$\hat{\boldsymbol{c}} \in R^n$，$\boldsymbol{x}(t) \in \boldsymbol{R}^n$是状态变量，$u(t) \in \boldsymbol{R}$是输入变量，$y(t) \in R$是输出变量。对于非奇异矩阵$\widehat{\boldsymbol{A}}$，将上述系统的状态方程左右同时乘以$\widehat{\boldsymbol{A}}^{-1}$，就可以得到：

$$\begin{cases} \boldsymbol{E}\dfrac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{x}(t) + \boldsymbol{b}u(t) \\ y(t) = \boldsymbol{c}^T\boldsymbol{x}(t) \end{cases}$$

其中$\boldsymbol{E} = \widehat{\boldsymbol{A}}^{-1}\widehat{\boldsymbol{E}}$，$\mathbf{b} = \widehat{\boldsymbol{A}}^{-1}\widehat{\boldsymbol{b}}$，$\mathbf{c} = \hat{\boldsymbol{c}}$。假设此系统初始状态为零，经过 Laplace 变换得到传递函数为$\boldsymbol{H}(s) = \boldsymbol{c}^T(s\boldsymbol{E} - \boldsymbol{I})^{-1}\boldsymbol{b}$，在s = 0处的第k阶矩是$m_k = -\boldsymbol{c}^T\boldsymbol{E}^k\boldsymbol{b}$。通过 Arnoldi 算法构造 Krylov 子空间$K_r(\boldsymbol{E};\boldsymbol{b})$的标准正交基$\boldsymbol{V} = [\boldsymbol{v}_1\ ,\boldsymbol{v}_2,\cdots\boldsymbol{v}_r]$，其中$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{E}\boldsymbol{V} = \boldsymbol{H}$，$\boldsymbol{H} = (\boldsymbol{h}_{ij}) \in \boldsymbol{R}^{r\times r}$是相应的上 Hessenberg 矩阵。

当计算出标准正交基后就可以由原始系统得到降阶系统

$$\begin{cases} \boldsymbol{V}^T\boldsymbol{E}\boldsymbol{V}\dfrac{d\tilde{\boldsymbol{x}}(t)}{dt} = \tilde{\boldsymbol{x}}(t) + \boldsymbol{V}^T\boldsymbol{b}u(t) \\ \tilde{y}(t) = \boldsymbol{c}^T\boldsymbol{V}\tilde{\boldsymbol{x}}(t) \end{cases}$$

其中$\tilde{\boldsymbol{x}}(t) \in \boldsymbol{R}^r$，$r$远小于$n$。又因为$\boldsymbol{V}^T\boldsymbol{E}\boldsymbol{V} = \boldsymbol{H}$，降阶系统可以变为

$$\begin{cases} \boldsymbol{H}\dfrac{d\tilde{\boldsymbol{x}}(t)}{dt} = \tilde{\boldsymbol{x}}(t) + \boldsymbol{V}^T\boldsymbol{b}u(t) \\ \tilde{y}(t) = \boldsymbol{c}^T\boldsymbol{V}\tilde{\boldsymbol{x}}(t) \end{cases}$$

原始系统完成降阶，转化为一个阶数相较以前小得多的系统。降阶系统传递函数的矩是$\tilde{m}_{\mathrm{k}} = -\boldsymbol{c}^T\boldsymbol{V}\boldsymbol{H}^k\boldsymbol{V}^T\boldsymbol{b}$，并且

$$\tilde{m}_k\boldsymbol{c}^T\boldsymbol{V}\boldsymbol{H}^k\boldsymbol{V}^T\boldsymbol{b} = -\|\boldsymbol{b}\|_2\boldsymbol{c}^T\boldsymbol{V}\boldsymbol{H}^k\boldsymbol{V}^T\frac{\boldsymbol{b}}{\|\boldsymbol{b}\|_2} = -\|\boldsymbol{b}\|_2\boldsymbol{c}^T\boldsymbol{V}\boldsymbol{H}^k\mathbf{e}_1 = -\boldsymbol{c}^T\boldsymbol{E}^k\boldsymbol{b} = m_k$$

其中的$k = 0$，1，$\cdots r-1$。所以 Arnoldi 降阶方法可以使降阶系统保持原系统

的前r阶矩。

（2）块 Arnoldi 算法

块 Arnoldi 算法是由普通的 Arnoldi 过程推广而来，它能够处理多输入多输出系统的模型降阶问题。具体的降阶过程与单输入单输出系统的降阶过程基本是一致的。以下面的多输入多输出系统为例

$$\begin{cases} \boldsymbol{E}\dfrac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \end{cases}$$

其中$\boldsymbol{E}$，$\boldsymbol{A} \in \boldsymbol{R}^{n\times n}$，$\boldsymbol{B} \in \boldsymbol{R}^{n\times p}$，$\boldsymbol{C} \in \boldsymbol{R}^{m\times n}$，$\boldsymbol{x}(t) \in \boldsymbol{R}^{n}$是状态变量，$\boldsymbol{u}(t) \in \boldsymbol{R}^{p}$是输入变量，$\boldsymbol{y}(t) \in \boldsymbol{R}^{m}$是输出变量。当矩阵$\boldsymbol{A}$为非奇异时，对该系统左右同时左乘$\boldsymbol{A}^{-1}$，可以得到

$$\begin{cases} \boldsymbol{G}\dfrac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{x}(t) + \boldsymbol{Q}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \end{cases}$$

其中$\boldsymbol{G} = \boldsymbol{A}^{-1}\boldsymbol{E}$，$\boldsymbol{Q} = \boldsymbol{A}^{-1}\boldsymbol{B}$。如果上面系统的初始状态为零，那么系统的传递函数$\boldsymbol{H}(s) = \boldsymbol{C}(s\boldsymbol{G} - \boldsymbol{I})^{-1}\boldsymbol{Q}$。由此可得传递函数$\boldsymbol{H}(s)$在$s = 0$处的第$i$阶矩为$M_i = -\boldsymbol{C}\boldsymbol{G}^{i}\boldsymbol{Q}$。对于多输入多输出系统来说，对应的 Krylov 子空间是由一个矩阵起始的。我们一般这样定义块 Krylov 子空间。由矩阵$\boldsymbol{G} \in \boldsymbol{R}^{n\times n}$，$\boldsymbol{Q} \in \boldsymbol{R}^{n\times p}$生成的$r$阶块 Krylov 子空间为$\boldsymbol{K}_r(\boldsymbol{G};\boldsymbol{Q}) = colspan\{\boldsymbol{Q}, \boldsymbol{G}\boldsymbol{Q}, \cdots, \boldsymbol{G}^{r-1}\boldsymbol{Q}\}$。可以看出块 Krylov 子空间$\boldsymbol{K}_r(\boldsymbol{G};\boldsymbol{Q})$最多可以有相互独立的$rp$个列向量，所以$dim\{\boldsymbol{K}_r(\boldsymbol{G};\boldsymbol{Q})\} \le rp$，块 Arnoldi 算法就是构造块 Krylov 子空间$\boldsymbol{K}_r(\boldsymbol{G};\boldsymbol{Q})$一组标准正交基$\boldsymbol{V} = [\boldsymbol{V}_0 \quad \boldsymbol{V}_1 \quad \cdots \quad \boldsymbol{V}_{r-1}]$的过程。

算法 2.1 块 Arnoldi 算法：

1、对矩阵$\boldsymbol{Q}$进行 QR 分解，记为$\boldsymbol{Q} = \boldsymbol{V}_0\boldsymbol{T}$；

2、计算$\widehat{\boldsymbol{V}}_1 = \boldsymbol{G}\boldsymbol{V}_0 - \boldsymbol{V}_0\boldsymbol{H}_{00}$，其中$\boldsymbol{H}_{00} = \boldsymbol{V}_0{}^{T}\boldsymbol{G}\boldsymbol{V}_0$，对$\widehat{\boldsymbol{V}}_1$进行 QR 分解，记为$\widehat{\boldsymbol{V}}_1 = \boldsymbol{V}_1\boldsymbol{H}_{10}$；

3、计算$\widehat{\boldsymbol{V}}_2 = \boldsymbol{G}\boldsymbol{V}_1 - \boldsymbol{V}_1\boldsymbol{H}_{11} - \boldsymbol{V}_0\boldsymbol{H}_0$，其中$\boldsymbol{H}_{11} = \boldsymbol{V}_1{}^{T}\boldsymbol{G}\boldsymbol{V}_1$，$\boldsymbol{H}_{01} = \boldsymbol{V}_0{}^{T}\boldsymbol{G}\boldsymbol{V}_1$。对$\widehat{\boldsymbol{V}}_2$进行 QR 分解，记为$\widehat{\boldsymbol{V}}_2 = \boldsymbol{V}_2\boldsymbol{H}_{21}$；

4、逐个进行下去，计算

$\widehat{\boldsymbol{V}}_r = \boldsymbol{G}\boldsymbol{V}_{r-1} - \boldsymbol{V}_{r-1}\boldsymbol{H}_{r-1,r-1} - \cdots - \boldsymbol{V}_1\boldsymbol{H}_{1,r-1} - \boldsymbol{V}_0\boldsymbol{H}_{0,r-1}$，其中$\boldsymbol{H}_{r-1,r-1} = \boldsymbol{V}_{r-1}{}^{T}\boldsymbol{G}\boldsymbol{V}_{r-1}$，$\ldots$，$\boldsymbol{H}_{1,r-1} = \boldsymbol{V}_1{}^{T}\boldsymbol{G}\boldsymbol{V}_{r-1}$，$\boldsymbol{H}_{0,r-1} = \boldsymbol{V}_0{}^{T}\boldsymbol{G}\boldsymbol{V}_{r-1}$。对$\widehat{\boldsymbol{V}}_r$进行 QR 分解，记为$\widehat{\boldsymbol{V}}_r = \boldsymbol{V}_r\boldsymbol{H}_{r,\,r-1}$。

可以看出，通过块 Arnoldi 算法可以得到标准正交矩阵$\boldsymbol{V}=[\boldsymbol{V}_0 \quad \boldsymbol{V}_1 \quad \cdots \quad \boldsymbol{V}_{r-1}]$和块上 Hessenberg 矩阵$\boldsymbol{H}=(\boldsymbol{H}_{ij})(i，j=0，1，\cdots，r-1)$。块 Arnoldi 算法具有和一般的 Arnoldi 过程相类似的性质：

$$colspan\{\boldsymbol{V}\}=\boldsymbol{K}_r(\boldsymbol{G};\boldsymbol{Q})，\quad \boldsymbol{V}^T\boldsymbol{G}\boldsymbol{V}=\boldsymbol{H}，\quad \boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}$$

在得到变换矩阵$\boldsymbol{V}\in\boldsymbol{R}^{n\times rp}$以后，就可以由原始系统得到其降阶系统

$$\begin{cases}\boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{E}\boldsymbol{V}\dfrac{d\tilde{\boldsymbol{x}}(t)}{dt}=\tilde{\boldsymbol{x}}(t)+\boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}(t)\\ \tilde{\boldsymbol{y}}(t)=\boldsymbol{C}\boldsymbol{V}\tilde{\boldsymbol{x}}(t)\end{cases}$$

其中$\tilde{\boldsymbol{x}}(t)\in\boldsymbol{R}^{rp}$，$rp$远小于$n$，那么上面的式子可以变为

$$\begin{cases}\boldsymbol{H}\dfrac{d\tilde{\boldsymbol{x}}(t)}{dt}=\tilde{\boldsymbol{x}}(t)+\boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{u}(t)\\ \tilde{\boldsymbol{y}}(t)=\boldsymbol{C}\boldsymbol{V}\tilde{\boldsymbol{x}}(t)\end{cases}$$

这个系统的传递函数是$\tilde{\boldsymbol{H}}(s)=\boldsymbol{C}\boldsymbol{V}(s\boldsymbol{H}-\boldsymbol{I})^{-1}\boldsymbol{V}^T\boldsymbol{Q}$，传递函数的第$i$阶矩为$\tilde{M}_i=-\boldsymbol{C}\boldsymbol{V}\boldsymbol{H}^i\boldsymbol{V}^T\boldsymbol{Q}$，该降阶系统的传递函数能够匹配原始系统的前$r$阶矩。

2.1.2 PRIMA方法

基于 Arnoldi 和 Lanczos 正交化过程的降阶方法通过匹配系统的传递函数的矩可以得到精度比较高的降阶模型，其实保持传递函数的矩与具体的 Arnoldi 和 Lanczos 正交化过程无关，应用一般 Krylov 子空间方法进行模型降阶的关键是要构造合适的变换矩阵，使变换矩阵的列向量所张成的空间能够包含在恰当的 Krylov 子空间中，降阶系统的矩依然能够得到保持。但是，对一些具有比较明显的特征和特殊结构的系统，在进行 Krylov 子空间降阶时需要考虑更多的因素。

Krylov 子空间模型降阶方法在满足特定条件时能够保持原始系统的稳定性和无源性。基于块 Arnoldi 过程的 PRIMA（Passive Reduced-order Interconnect Macromodeling Algorithm）降阶方法[45]因系统的特殊形式而得以保持其无源性。

现代高速集成电路系统中大量互连线系统的等价电路系统经过标准的节点分析后具有如下形式

$$\begin{cases}\boldsymbol{E}\dfrac{d\boldsymbol{x}(t)}{dt}=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)\\ \boldsymbol{y}(t)=\boldsymbol{B}^T\boldsymbol{x}(t)\end{cases}$$

其中矩阵$\boldsymbol{E}$，$\boldsymbol{A}$和$\boldsymbol{B}$具有如下形式

$$\boldsymbol{E}=\begin{bmatrix}\boldsymbol{M} & 0\\ 0 & \boldsymbol{H}\end{bmatrix}\in\boldsymbol{R}^{n\times n}，\quad \boldsymbol{A}=-\begin{bmatrix}\boldsymbol{N} & \boldsymbol{F}\\ -\boldsymbol{F}^T & 0\end{bmatrix}\in\mathrm{R}^{\mathrm{n}\times\mathrm{n}}，\quad \boldsymbol{B}=\begin{bmatrix}\boldsymbol{B}_1\\ 0\end{bmatrix}\in\boldsymbol{R}^{n\times p}$$

上面的矩阵$\boldsymbol{M}$，$\boldsymbol{H}$和$\boldsymbol{N}$为半正定矩阵。对块 Krylov 子空间$\mathbf{K}_{\mathrm{r}}(\mathbf{A}^{-1}\boldsymbol{E};\boldsymbol{A}^{-1}\boldsymbol{B})$实施块 Arnoldi 过程，可以得到标准正交矩阵$\boldsymbol{V}\in\boldsymbol{R}^{n\times rp}$（其中$rp$远小于$n$）。这样，就可以根据变换矩阵$\boldsymbol{V}$得到原始系统的降阶系统如下

$$\begin{cases}\widetilde{\boldsymbol{E}}\dfrac{d\widetilde{\boldsymbol{x}}(t)}{dt}=\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}}(t)+\widetilde{\boldsymbol{B}}\boldsymbol{u}(t)\\ \widetilde{\boldsymbol{y}}(t)=\widetilde{\boldsymbol{B}}^{T}\widetilde{\boldsymbol{x}}(t)\end{cases}$$

其中$\widetilde{\boldsymbol{x}}(t)\in\mathbf{R}^{\mathrm{rp}}$，$\widetilde{\boldsymbol{E}}=\boldsymbol{V}^{\mathrm{T}}\boldsymbol{E}\boldsymbol{V}$，$\widetilde{\boldsymbol{A}}=\boldsymbol{V}^{T}\boldsymbol{A}\boldsymbol{V}$，$\widetilde{\boldsymbol{B}}=\boldsymbol{V}^{T}\boldsymbol{B}$。上面的系统就是降阶系统，它的传递函数是$\widetilde{\boldsymbol{H}}(s)=\widetilde{\boldsymbol{B}}^{\mathrm{T}}\left(s\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\widetilde{\boldsymbol{B}}$。我们能够知道降阶系统能够匹配原系统的前$r$阶矩。这个过程一般称之为 PRIMA 降阶过程。

PRIMA 算法可以保持系统的无源性，这只需要验证降阶系统的传递函数$\widetilde{\boldsymbol{H}}(\mathrm{s})$满足下面的两个条件：（1）对任意的$\mathrm{s}\in\boldsymbol{C}$，成立$\widetilde{\boldsymbol{H}}(\bar{s})=\overline{\widetilde{\boldsymbol{H}}(s)}$；（2）对任意的实部大于零的$\mathrm{s}\in\mathbf{C}$以及任意向量$\boldsymbol{z}\in\boldsymbol{C}^{p}$，成立$\boldsymbol{z}^{H}(\widetilde{\boldsymbol{H}}(s)+\widetilde{\boldsymbol{H}}^{H}(s))\boldsymbol{z}\geq 0$。因为$\widetilde{\boldsymbol{E}}$，$\widetilde{\boldsymbol{A}}$，$\widetilde{\boldsymbol{B}}$和$\widetilde{\boldsymbol{C}}$为实矩阵，所以条件（1）自然就满足了。下面验证条件（2）。

令$\widehat{\boldsymbol{H}}(s)=\widetilde{\boldsymbol{H}}(s)+\widetilde{\boldsymbol{H}}^{H}(s)$，可以得到

$$\boldsymbol{z}^{H}\widehat{H}(s)\boldsymbol{z}=\boldsymbol{z}^{H}\left(\widetilde{\boldsymbol{B}}^{T}\left(s\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\widetilde{\boldsymbol{B}}+\widetilde{\boldsymbol{B}}^{T}\left(\bar{s}\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-T}\widetilde{\boldsymbol{B}}\right)\boldsymbol{z}$$

$$=\boldsymbol{z}^{H}\widetilde{\boldsymbol{B}}^{T}(\left(s\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}+\left(\bar{s}\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-T})\widetilde{\boldsymbol{B}}\boldsymbol{z}$$

$$=\boldsymbol{z}^{H}\widetilde{\boldsymbol{B}}^{T}\left(s\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\left(\left(\bar{s}\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)+\left(\bar{s}\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{T}\right)\left(\bar{s}\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-T}\widetilde{\boldsymbol{B}}\boldsymbol{z}$$

将ω记做$\left(\bar{s}\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-T}\widetilde{\boldsymbol{B}}\boldsymbol{z}$，令$s=a+ib$，上式可以化简为

$$\boldsymbol{z}^{H}\widehat{H}(s)\boldsymbol{z}=\omega^{H}(\left((a+ib)\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)+((a-ib)\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}})^{T})\omega$$

$$=\omega^{H}\left(a\left(\widetilde{\boldsymbol{E}}+\widetilde{\boldsymbol{E}}^{T}\right)-\widetilde{\boldsymbol{A}}-\widetilde{\boldsymbol{A}}^{T}\right)\omega$$

$$=\omega^{H}(a(\boldsymbol{V}^{T}\boldsymbol{E}\boldsymbol{V}+\boldsymbol{V}^{T}\boldsymbol{V}\boldsymbol{E}^{T}\boldsymbol{V})-\boldsymbol{V}^{T}\boldsymbol{A}\boldsymbol{V}-\boldsymbol{V}^{T}\boldsymbol{A}^{T}\boldsymbol{V})\omega$$

$$=\omega^{H}\boldsymbol{V}^{T}(a(\boldsymbol{E}+\boldsymbol{E}^{T})-\boldsymbol{A}-\boldsymbol{A}^{T})\boldsymbol{V}\omega$$

再记$\boldsymbol{f}=\boldsymbol{V}\omega$，由上式可以得到$\boldsymbol{z}^{H}\widehat{\boldsymbol{H}}(s)\boldsymbol{z}=\boldsymbol{f}^{H}(a(\boldsymbol{E}+\boldsymbol{E}^{T})-\boldsymbol{A}-\boldsymbol{A}^{T})\boldsymbol{f}$。

要证明矩阵$\widehat{\boldsymbol{H}}(s)$是半正定的，现在只需证明矩阵$a(\boldsymbol{E}+\boldsymbol{E}^{T})-\boldsymbol{A}-\boldsymbol{A}^{T}$是半正定的即可。由于$\boldsymbol{E}$为半正定矩阵，因此对$a=Re\{s\}>0$均有$\boldsymbol{f}\boldsymbol{H}a(\boldsymbol{E}+\boldsymbol{E}^{T})\boldsymbol{f}\geq 0$。可以注意到矩阵$\boldsymbol{A}$的特殊结构，并且矩阵$\boldsymbol{N}$为半正定矩阵，因此对任意复向量$\boldsymbol{f}$，有下式成立

$$\boldsymbol{f}^H(-\boldsymbol{A}-\boldsymbol{A}^T)\boldsymbol{f}=\boldsymbol{f}^H\left(\begin{bmatrix}\boldsymbol{N} & \boldsymbol{F}\\ -\boldsymbol{F}^T & 0\end{bmatrix}+\begin{bmatrix}\boldsymbol{N} & \boldsymbol{F}\\ -\boldsymbol{F}^T & 0\end{bmatrix}^T\right)\boldsymbol{f}$$
$$=\boldsymbol{f}^H\begin{bmatrix}2\boldsymbol{N} & 0\\ 0 & 0\end{bmatrix}\boldsymbol{f}\geq 0$$

上式表明$\widehat{\boldsymbol{H}}(s)$是半正定的，即条件（2）成立。

若线性时不变系统是无源的，则它必是稳定的。两个稳定的线性时不变系统连接在一起后并不一定能保持系统的稳定性，但两个无源线性时不变系统互连后系统仍是无源的，因此也是稳定的。PRIMA 算法因保持系统的无源性而在实际应用中备受关注。

2.1.3 SPRIM方法

PRIMA 算法[47]因为充分考虑了系统的结构特征而保持了系统的无源性，但是，应用 PRIMA 算法所得到的降阶系统会破坏原始系统系数矩阵的某些良好结构。

再考虑系统（2. 9），系数矩阵$\boldsymbol{A}$和$\boldsymbol{E}$的分块结构对实际系统的实现具有重要意义，从而，也希望降阶系统（2. 10）的系数矩阵$\tilde{A}$和$\tilde{E}$仍然具有类似结构。SPRIM（Structure-Preserving Reduced-order Interconnect Macromodeling）算法对 PRIMA 算法进行了改进，它能够保持系统（2. 9）的结构特征，并且具有更高的计算精度。

易知，在 Krylov 子空间降阶方法中，若变换矩阵$\boldsymbol{V}$满足

$$\text{colspan}\{\boldsymbol{V}\}=\boldsymbol{K}_r(\boldsymbol{A}^{-1}\boldsymbol{E};\boldsymbol{A}^{-1}\boldsymbol{B})$$

则降阶系统（2. 10）可以匹配原始系统前 r 阶矩。进一步，若矩阵$\boldsymbol{V}$满足

$$\boldsymbol{K}_r(\boldsymbol{A}^{-1}\boldsymbol{E};\boldsymbol{A}^{-1}\boldsymbol{B})\subseteq\text{colspan}\{\boldsymbol{V}\}$$

现在假设$\widetilde{\mathbf{V}}$为由 PRIMA 算法得到的变换矩阵，满足

$$\text{colspan}\{\widetilde{\mathbf{V}}\}=\boldsymbol{K}_{\mathrm{r}}(\boldsymbol{A}^{-1}\boldsymbol{E};\boldsymbol{A}^{-1}\boldsymbol{B})$$

对矩阵$\widetilde{\mathbf{V}}$按照系数矩阵$\boldsymbol{E}$和$\boldsymbol{A}$的块结构做划分$\widetilde{\mathbf{V}}=[\boldsymbol{V}_1{}^T\quad \boldsymbol{V}_2{}^T]^T$，然后令$\boldsymbol{V}=\text{diag}\{\boldsymbol{V}_1,\boldsymbol{V}_2\}$，容易明白

$$\boldsymbol{K}_r(\boldsymbol{A}^{-1}\boldsymbol{E};\boldsymbol{A}^{-1}\boldsymbol{B})=\text{colspan}\{\widetilde{\boldsymbol{V}}\}\subseteq\text{colspan}\{\boldsymbol{V}\}$$

根据变换矩阵$\boldsymbol{V}\in\mathbf{R}^{\mathrm{n}\times 2\mathrm{rp}}(\mathrm{rp}\ll\mathrm{n})$，就可以得到原始系统（2. 9）的降阶系统。变换矩阵$\boldsymbol{V}$的块对角结构使得原始系统（2. 9）的结构特征得以保持。

算法 2.2（SPRIM 算法）

第一步 给定原始系统（2. 9）的稀疏矩阵

$$\boldsymbol{E}=\begin{bmatrix}\boldsymbol{M} & 0\\ 0 & \boldsymbol{H}\end{bmatrix}\in \mathbf{R}^{n\times n},\quad \boldsymbol{A}=-\begin{bmatrix}\boldsymbol{N} & \boldsymbol{F}\\ -\boldsymbol{F}^{T} & 0\end{bmatrix}\in \mathbf{R}^{n\times n},\quad \boldsymbol{B}=\begin{bmatrix}\boldsymbol{B}_1\\ 0\end{bmatrix}\in \mathbf{R}^{n\times p}$$

其中$\boldsymbol{M}$，$\boldsymbol{N}$和$\boldsymbol{B}_1$具有相同的行数，$\boldsymbol{M}$，$\boldsymbol{H}$和$\boldsymbol{N}$均为半正定矩阵。同时，给定展开点s_0，以及正整数r。

第二步 令$\boldsymbol{G}=(\mathrm{s}_0\boldsymbol{E}-\boldsymbol{A})^{-1}\boldsymbol{E}$，$\boldsymbol{Q}=(\mathrm{s}_0\boldsymbol{E}-\boldsymbol{A})^{-1}\boldsymbol{B}$，应用块 Arnoldi 算法计算矩阵$\widetilde{\boldsymbol{V}}$，使其满足colspan$\{\widetilde{\boldsymbol{V}}\}=\boldsymbol{K}_{\mathrm{r}}(\boldsymbol{G};\boldsymbol{Q})$

第三步 根据矩阵$\boldsymbol{E}$和$\boldsymbol{A}$的块结构特征对矩阵$\widetilde{\boldsymbol{V}}$做划分$\widetilde{\boldsymbol{V}}=[\boldsymbol{V}_1{}^{T}\quad \boldsymbol{V}_2{}^{T}]^{T}$，并且构造相应的矩阵$\boldsymbol{V}=\mathrm{diag}\{\boldsymbol{V}_1,\boldsymbol{V}_2\}$。

第四步 令$\widetilde{\boldsymbol{M}}=\boldsymbol{V}_1{}^{T}\boldsymbol{M}\boldsymbol{V}_1$，$\widetilde{\boldsymbol{H}}=\boldsymbol{V}_2{}^{T}\boldsymbol{H}\boldsymbol{V}_2$，$\widetilde{\boldsymbol{N}}=\boldsymbol{V}_1{}^{T}\boldsymbol{N}\boldsymbol{V}_1$，$\widetilde{\boldsymbol{F}}=\boldsymbol{V}_1{}^{T}\boldsymbol{F}\boldsymbol{V}_2$，$\widetilde{\boldsymbol{B}}_1=\boldsymbol{V}_1{}^{T}\boldsymbol{B}_1$，建立原始系统（2. 9）的降阶系统为

$$\begin{cases}\widetilde{\boldsymbol{E}}\dfrac{d\tilde{x}(t)}{dt}=\widetilde{\boldsymbol{A}}\tilde{x}(t)+\widetilde{\boldsymbol{B}}u(t)\\ \quad\tilde{y}(t)=\widetilde{\boldsymbol{B}}^{T}\tilde{x}(t)\end{cases}$$

其中

$$\widetilde{\boldsymbol{E}}=\begin{bmatrix}\widetilde{\boldsymbol{M}} & 0\\ 0 & \widetilde{\boldsymbol{H}}\end{bmatrix},\quad \boldsymbol{A}=-\begin{bmatrix}\widetilde{\boldsymbol{N}} & \widetilde{\boldsymbol{F}}\\ -\widetilde{\boldsymbol{F}}^{T} & 0\end{bmatrix},\quad \widetilde{\boldsymbol{B}}=\begin{bmatrix}\widetilde{\boldsymbol{B}}_1\\ 0\end{bmatrix}$$

由上述算法的第四步可知，与上小节的降阶系统（2. 10）不同，这里的降阶系统（2. 11）保持了原始系统（2. 9）的系数矩阵的结构特征。

应用 SPRIM 算法所得到的降阶系统的传递函数为$\widetilde{\boldsymbol{H}}(s)=\widetilde{\boldsymbol{B}}^{T}\left(s\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\widetilde{\boldsymbol{B}}$，其中系数矩阵$\widetilde{\boldsymbol{E}}$和$\widetilde{\boldsymbol{A}}$均为$2rp$阶方阵。考虑到矩阵$\widetilde{\boldsymbol{E}}$，$\widetilde{\boldsymbol{A}}$和$\widetilde{\boldsymbol{B}}$的特殊结构，可以计算得到

$$\left(s\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}=\begin{bmatrix}\left(s\widetilde{\boldsymbol{M}}+\widetilde{\boldsymbol{N}}+s^{-1}\widetilde{\boldsymbol{F}}\widetilde{\boldsymbol{H}}^{-1}\widetilde{\boldsymbol{F}}^{T}\right)^{-1} & *\\ * & *\end{bmatrix}$$

因此，降阶系统（2. 11）的传递函数为$\widetilde{\boldsymbol{H}}(s)=\widetilde{\boldsymbol{B}}1^{T}\left(s\widetilde{\boldsymbol{M}}+\widetilde{\boldsymbol{N}}+s^{-1}\widetilde{\boldsymbol{F}}\widetilde{\boldsymbol{H}}^{-1}\widetilde{\boldsymbol{F}}^{T}\right)^{-1}\widetilde{\boldsymbol{B}}_1$，其中矩阵$\widetilde{\boldsymbol{M}}$，$\widetilde{\boldsymbol{N}}$和$\widetilde{\boldsymbol{F}}\widetilde{\boldsymbol{H}}^{-1}\widetilde{\boldsymbol{F}}^{\mathrm{T}}$均为$rp$阶方阵，$\widetilde{\boldsymbol{B}}1$为$rp\times p$矩阵。一次降阶系统可以转化为$rp$阶系统。

由于 PRIMA 算法保持系统无源性与变换矩阵$\boldsymbol{V}$的具体选择无关，所以 SPRIM 算法也能保持系统的无源性。此外，较 PRIMA 算法而言，由 SPRIM 算法所得降阶系统能匹配原始系统更多阶矩，这由下面定理可以看出。

定理 2.1 若$\mathrm{s}_0\in\mathrm{R}$，变换矩阵$\boldsymbol{V}$是由算法 2.2 得到的，则降阶系统（2. 11）的传递函数$\widetilde{\boldsymbol{H}}(\mathrm{s})$与原始系统（2. 9）的传递函数$\boldsymbol{H}(\mathrm{s})$在$s_0$处具有相同的前$2r$阶矩。

证明：注意到，原始系统（2. 9）的传递函数为

$$\boldsymbol{H}(s)=\boldsymbol{B}^T(s\boldsymbol{E}-\boldsymbol{A})^{-1}\boldsymbol{B}=\sum_{k=0}^{+\infty}(-1)^k\boldsymbol{B}^T\boldsymbol{G}^k\boldsymbol{Q}(s-s_0)^k$$

类似地，对降阶系统（2. 11）的传递函数$\widetilde{\boldsymbol{H}}(s)=\widetilde{\boldsymbol{B}}^T\left(s\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\widetilde{\boldsymbol{B}}$，也可以得到$\widetilde{\boldsymbol{H}}(s)=\sum_{k=0}^{+\infty}(-1)^k\widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{G}}^k\widetilde{\boldsymbol{Q}}(s-s_0)^k$，其中$\widetilde{\boldsymbol{G}}=\left(s_0\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\widetilde{\boldsymbol{E}}$，$\widetilde{\boldsymbol{Q}}=\left(s_0\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\widetilde{\boldsymbol{B}}$。为证明定理结论成立，只需证明对任意$i=0，1，\cdots，2r-1$，式$\widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{G}}^k\widetilde{\boldsymbol{Q}}=\boldsymbol{B}^T\boldsymbol{G}^k\boldsymbol{Q}$成立。

首先，对任意$j=0，1，\cdots，r-1$，由于$colspan\{\boldsymbol{G}^j\boldsymbol{Q}\}\subseteq\boldsymbol{K}_r(\boldsymbol{G};\boldsymbol{Q})$，因而存在矩阵$\boldsymbol{A}_j$满足$\boldsymbol{G}^j\boldsymbol{Q}=\boldsymbol{V}\boldsymbol{\Lambda}_j$。又注意到

$$\boldsymbol{Q}=\left(s_0\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{V}^T(s_0\boldsymbol{E}-\boldsymbol{A})(s_0\boldsymbol{E}-\boldsymbol{A})^{-1}\boldsymbol{B}$$

$$=\left(s_0\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{V}^T(s_0\boldsymbol{E}-\boldsymbol{A})\boldsymbol{V}\boldsymbol{\Lambda}_0=\boldsymbol{\Lambda}_0$$

应用上面的关系可以推得

$\widetilde{\boldsymbol{G}}\widetilde{\boldsymbol{Q}}$

$=(s_0\widetilde{\boldsymbol{E}}$

$-\widetilde{\boldsymbol{A}})^{-1}\boldsymbol{V}^T\boldsymbol{E}\boldsymbol{V}\boldsymbol{\Lambda}_0$

$$=\left(s_0\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{V}^T(s_0\boldsymbol{E}-\boldsymbol{A})(s_0\boldsymbol{E}-\boldsymbol{A})^{-1}\boldsymbol{E}(s_0\boldsymbol{E}-\boldsymbol{A})^{-1}\boldsymbol{B}$$

$$=\left(s_0\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{V}^T(s_0\boldsymbol{E}-\boldsymbol{A})\boldsymbol{V}\boldsymbol{\Lambda}_1=\boldsymbol{\Lambda}_1$$

再用归纳法可以证明对$j=0，1，\cdots，r-1$，式$\widetilde{\boldsymbol{G}}_j\widetilde{\boldsymbol{Q}}=\boldsymbol{\Lambda}_j$成立。进而，总有

$$\boldsymbol{V}\widetilde{\boldsymbol{G}}^j\widetilde{\boldsymbol{Q}}=\boldsymbol{G}^j\boldsymbol{Q}，\ j=0，1，\cdots，r-1$$

其次，注意到系数矩阵$\boldsymbol{E}$，$\boldsymbol{A}$和$\boldsymbol{B}$的特殊结构，不难验证对$i=0，1，\cdots，r-1$，下式成立

$$(\boldsymbol{G}^T)^i=(\boldsymbol{E}(s_0\boldsymbol{J}\boldsymbol{E}\boldsymbol{J}-\boldsymbol{J}\boldsymbol{A}\boldsymbol{J})^{-1})^i=\boldsymbol{J}^{-1}(\boldsymbol{E}(s_0\boldsymbol{E}-\boldsymbol{A})^{-1})^i\boldsymbol{J}，\ \boldsymbol{B}=\boldsymbol{J}\boldsymbol{B}$$

其中$\boldsymbol{J}=\boldsymbol{J}^{-1}=\mathrm{diag}\{\boldsymbol{I}_1，\ -\boldsymbol{I}_2\}$与$\boldsymbol{E}$和$\boldsymbol{A}$具有相同的块结构。类似地，还有

$$\left(\widetilde{\boldsymbol{G}}^T\right)^i=\left(\widetilde{\boldsymbol{E}}\left(s_0\boldsymbol{J}_r\widetilde{\boldsymbol{E}}\boldsymbol{J}_r-\boldsymbol{J}_r\widetilde{\boldsymbol{A}}\boldsymbol{J}_r\right)^{-1}\right)^i=\boldsymbol{J}_r{}^{-1}\left(\widetilde{\boldsymbol{E}}\left(s_0\widetilde{\boldsymbol{E}}-\widetilde{\boldsymbol{A}}\right)^{-1}\right)^i\boldsymbol{J}_r\qquad(2.13)$$

其中$\boldsymbol{J}_r=\boldsymbol{J}_r{}^{-1}=diag\{\boldsymbol{I}_{r1}，\ -\boldsymbol{I}_{r2}\}$与$\boldsymbol{E}$和$\boldsymbol{A}$具有相同的块结构。再注意到$\boldsymbol{B}=\boldsymbol{J}\boldsymbol{B}$，从而有

$$(\boldsymbol{G}^T)^i\boldsymbol{B}=\boldsymbol{J}^{-1}(\boldsymbol{E}(s_0\boldsymbol{E}-\boldsymbol{A})^{-1})^i\boldsymbol{J}\boldsymbol{B}=\boldsymbol{J}^{-1}(\boldsymbol{E}(s_0\boldsymbol{E}-\boldsymbol{A})^{-1})^i(s_0\boldsymbol{E}-\boldsymbol{A})\boldsymbol{Q}$$

$$=\boldsymbol{J}^{-1}\boldsymbol{E}\boldsymbol{G}^{i-1}\boldsymbol{Q}$$

再将式（2. 12）代入上式，得到

$$(\boldsymbol{G}^T)^i\boldsymbol{B} = \boldsymbol{J}^{-1}\boldsymbol{E}\boldsymbol{V}\widetilde{\boldsymbol{G}}^{i-1}\widetilde{\boldsymbol{Q}} = \boldsymbol{J}^{-1}\boldsymbol{E}\boldsymbol{V}\widetilde{\boldsymbol{G}}^{i-1}\left(s_0\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)^{-1}\widetilde{\boldsymbol{B}}$$

进一步，由前面式子还有$\boldsymbol{B}^T\boldsymbol{G}^i = \widetilde{\boldsymbol{B}}^T\left(s_0\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)^{-T}\left(\widetilde{\boldsymbol{G}}^T\right)^{i-1}\boldsymbol{V}^T\boldsymbol{E}\boldsymbol{J}^{-1}$,对该式两端右乘$\boldsymbol{V}$，得到

$$\boldsymbol{B}^T\boldsymbol{G}^i\boldsymbol{V} = \widetilde{\boldsymbol{B}}^T\left(s_0\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)^{-T}\left(\widetilde{\boldsymbol{G}}^T\right)^{i-1}\boldsymbol{V}^T\boldsymbol{E}\boldsymbol{J}^{-1}\boldsymbol{V}$$

再将（2. 13）代入上式，就有

$$\boldsymbol{B}^T\boldsymbol{G}^i\boldsymbol{V} = \widetilde{\boldsymbol{B}}^T\boldsymbol{J}_r\left(s_0\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)^{-T}\mathbf{J}_r{}^{-1}\left(\widetilde{\boldsymbol{E}}\left(s_0\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)^{-1}\right)^{i-1}\boldsymbol{J}_r\boldsymbol{V}^T\boldsymbol{E}\boldsymbol{J}^{-1}\boldsymbol{V}$$

$$= \widetilde{\boldsymbol{B}}^T\left(s_0\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)^{-1}\left(\widetilde{\boldsymbol{E}}\left(s_0\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)^{-1}\right)^{i-1}\widetilde{\boldsymbol{E}}$$

$$= \widetilde{\boldsymbol{B}}^T\left(\left(s_0\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)^{-1}\widetilde{\boldsymbol{E}}\right)^i = \widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{G}}^i,\ i = 0,\ 1,\ \cdots,\ r$$

现在，对任意$k \in \{0,\ 1,\ \cdots,\ 2r-1\}$，它可以表示为$k = i + j$，其中$i \in \{0,\ 1,\ \cdots,\ r\}$，$j \in \{0,\ 1,\ \cdots,\ r-1\}$。这样，结合式（2. 12）和式（2. 14），就有

$$\widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{G}}^k\widetilde{\boldsymbol{Q}} = \widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{G}}^i\widetilde{\boldsymbol{G}}^j\widetilde{\boldsymbol{Q}} = \boldsymbol{B}^T\boldsymbol{G}^i\boldsymbol{V}\widetilde{\boldsymbol{G}}^j\widetilde{\boldsymbol{Q}} = \boldsymbol{B}^T\boldsymbol{G}^i\boldsymbol{G}^j\boldsymbol{Q} = \boldsymbol{B}^T\boldsymbol{G}^k\boldsymbol{Q}$$

上式表明定理结论成立，证毕。

2.1.4 PMTBR方法

Krylov 子空间投影方法在实际应用中有两个缺点。首先，如何控制这些方法中的误差没有的公认的标准。虽然存在一些估算误差的方法[3]，但他们在实践中很少使用。这些估算方法的缺点是，他们需要额外的计算，这可能是代价很高而且很难实现的，并且过程中也会产生估计误差。模型降阶方法的一个类别是平衡截断法，这些声称“几乎最佳”的方法，可以很方便地计算后验误差范围。 但是直接适用于集成电路问题的 TBR 的方法代价过于昂贵，也有人提出了结合平衡截断和 Krylov 子空间投影的方法。这些混合 Krylov 子空间投影和 TBR 的方法的确取得了不错的效果，但是它们在处理很不平衡的非对称系统时必须结合两个单独的投影子空间，这样实际实施起来就太过复杂，所以还没有被广泛用于处理实际的问题和工程实践。

假设一个简单的系统$\boldsymbol{A} = \boldsymbol{A}^T$，$\boldsymbol{C} = \boldsymbol{B}^T$，并且$\boldsymbol{A}$是正定的。这可以看作一个 RC 网络经过分析后形成的系统，已知标准的 TBR 算法会产生被动近似。可以看出在对称的例子中两个格拉姆行列式是匹配的，标准 TBR 算法是通过求解李雅普诺夫方程实现的。

格拉姆行列式X也可以叫做微分方程$\frac{dx}{dt} = \boldsymbol{Ax} + \boldsymbol{Bu}$的基本解。时域的格拉姆行列式可以用下式计算

$$\mathrm{X} = \int_0^\infty \boldsymbol{e^{At}BB^T e^{A^T t}} dt$$

另外，$\boldsymbol{e^{At}}$的拉普拉斯变换是$(s\boldsymbol{I} - \boldsymbol{A})^{-1}$，那么格拉姆行列式也可以通过下面的式子来计算得到

$$\boldsymbol{X} = \int_{-\infty}^{\infty} (j\omega \boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{BB}^T (j\omega \boldsymbol{I} - \boldsymbol{A})^{-H} d\omega$$

其中上标H表示 Hermitian 转置，考虑上式的数值积分。根据节点ω_k和权重w_k做正交，定义下式

$$\boldsymbol{z}_k = (j\omega_k \boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{B}$$

那么$\boldsymbol{X}$的近似$\widehat{\boldsymbol{X}}$可以由下式计算得到

$$\widehat{\boldsymbol{X}} = \sum_k \omega_k \boldsymbol{z}_k \boldsymbol{z}_k^H$$

以$\boldsymbol{z}_k$为列构造矩阵$\boldsymbol{Z}$，同时构造以$\boldsymbol{W}_{kK} = \sqrt{\omega_k}$为对角元的对角阵$\boldsymbol{W}$。上式可以简化为

$$\widehat{\boldsymbol{X}} = \boldsymbol{ZW}^2 \boldsymbol{Z}^H$$

为了推导模型降阶过程，考虑特征值分解

$$\boldsymbol{X} = \boldsymbol{V}_L \boldsymbol{\Sigma} \boldsymbol{V}_L^T$$

注意当$\boldsymbol{X}$是实对称阵时，$\boldsymbol{V}_L \boldsymbol{V}_L^T = \boldsymbol{I}$。显然，用于构造投影空间的$\boldsymbol{V}$的列向量相当于$\boldsymbol{X}$的主特征向量。如果求积是精确的，那么$\widehat{\boldsymbol{X}}$将收敛于$\boldsymbol{X}$。这意味这$\widehat{\boldsymbol{X}}$收敛于$\boldsymbol{X}$的特征空间。现在考虑$\boldsymbol{ZW}$的奇异值分解

$$\boldsymbol{ZW} = \boldsymbol{V}_Z \boldsymbol{S}_Z \boldsymbol{U}_Z$$

其中$\boldsymbol{S}_Z$是实对角阵，$\boldsymbol{V}_Z$和$\boldsymbol{U}_Z$是单位阵，显然

$$\widehat{\boldsymbol{X}} = \boldsymbol{V}_Z \boldsymbol{S}_Z^2 \boldsymbol{V}_Z^T$$

所以，$\boldsymbol{V}_Z$的奇异向量可以通过$\boldsymbol{S}_Z$的奇异值来确定，从而给出$\widehat{\boldsymbol{X}}$的特征向量。因此，$\boldsymbol{V}_Z$收敛到$\boldsymbol{X}$的特征空间，同时 Hankel 奇异值可以通过$\boldsymbol{S}_Z$直接得到。然后$\boldsymbol{V}_Z$可以用作降阶系统的投影矩阵。

看起来好像矩阵$\boldsymbol{Z}$的奇异值与近似误差有关，但是上面的阐述说明它其实是

准确的，矩阵$\boldsymbol{Z}$的奇异值分解与 TBR 说明了同样的问题。PMTBR 相比 TBR 算法，能够显著的降低问题的复杂度，同时拥有快速的处理速度。

算法 2.3 PMTBR 算法：

1. Do until satisfied:
2. Select a frequency point s_i.
3. Compute $z_i = [s_i I - A]^{-1} B$.
4. Form the matrix of columns
5. [real s_i]: $Z_{(i)} = [z_1, z_2, \ldots, z_i]$.
   [complex s_i]: $Z_{(i)} = [z_1, z_2, \ldots, z_i, z_i^*]$.
6. Construct the SVD of $Z_{(i)}$.
   If the error is satisfactory, set $Z = Z_{(i)}$, go to Step 7.
   Otherwise, go to Step 2.
7. Construct the projection space V from the orthogonalized column span of Z, dropping columns whose associated singular values fall below a desired tolerance.

2.2 应用于大规模 RC 网络的模型降阶方法

2.2.1 SIP算法

SIP 算法[53]主要是应对可以用稀疏矩阵表示的大规模动态系统，特别是具有大量输入、输出端口的系统。该算法结合了现金部分应用广泛的算法的优点，如 PRIMA 和 TICER，同时避免了两者的缺陷。该算法能都实现高阶逼近，同时利用系统的稀疏性从而保持系统的无源性。

从数学上来说，SIP 算法属于投影方法，并继承其准确性（通过多点拟合），一般性，同时能够保持无源性。然而，SIP 的实现基于稀疏矩阵操作，也类似于节点消去方法，所以效率更高，并且能够处理相对现有投影方法更大量级和规模的问题。事实上，SIP 实施可以直接基于高度优化的现成的稀疏矩阵，某种意义上来说是基于稀疏矩阵降阶方法，而在 PRIMA 等算法中其核心计算是基于密集矩阵。这意味，SIP 方法可以扩展到更复杂的问题，例如一般系统，非对称矩

阵系统，同时保留其基本的简单性和效率。SIP 算法与投影方法，消去方法，稀疏矩阵的高斯消去之间有密切的联系。SIP 在某些情况下会消减到 PRIMA，某些情况下，SIP 会减少到 SGE，并在某些情况下，SIP，SGE，TICER，和 PRIMA 会产生数学上等价的模型，同时 SIP 算法还能够有效的与其他算法相结合。

一个 RC 网络可以写成如下形式

$$\begin{cases} (\boldsymbol{G} + s\boldsymbol{C})\boldsymbol{x} = \boldsymbol{E}\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{E}^T \boldsymbol{x} \end{cases}$$

其中$\boldsymbol{u}$是输入向量（如端口电流），$\boldsymbol{x}$是内部网络节点电压，$\boldsymbol{Y}$是输出向量（如端口电压）。 $\boldsymbol{G}$，$\boldsymbol{C} \in \boldsymbol{R}^{n\times n}$代表电导矩阵和电容矩阵,$\boldsymbol{E} \in \boldsymbol{R}^{n\times m}$是一个端口映射到节点的拓扑矩阵。 $\boldsymbol{E}$可以写成

$$\boldsymbol{E} = \begin{bmatrix} 0 \\ \boldsymbol{I} \end{bmatrix}$$

假设通过节点置换后，所有的内部节点放置于前面，端口节点放置于最后。通常情况下，$\boldsymbol{C}$和$\boldsymbol{G}$是半正定并且对称的。

通过构造$\boldsymbol{M} \in \boldsymbol{R}^{n\times mq}$进行正交投影

$$\widehat{\boldsymbol{G}} = \boldsymbol{M}^T\boldsymbol{G}\boldsymbol{M}，\ \widehat{\boldsymbol{C}} = \boldsymbol{M}^T\boldsymbol{C}\boldsymbol{M}，\ \widehat{\boldsymbol{E}} = \boldsymbol{M}^T\boldsymbol{E}$$

那么原系统和降阶系统的传递函数可以写为

$$\boldsymbol{H}(s) = \boldsymbol{E}^T(\boldsymbol{G} + s\boldsymbol{C})^{-1}\boldsymbol{E}$$
$$\hat{H}(s) = \widehat{\boldsymbol{E}}^T\left(\widehat{\boldsymbol{G}} + s\widehat{\boldsymbol{C}}\right)^{-1}\widehat{\boldsymbol{E}}$$

可以看出，如果mq远小于n，那么系统的大小会比降阶前小得多。

PRIMA 算法需要将矩阵$\boldsymbol{M}$显式的构造出来，而且$\boldsymbol{M}$是稠密的，同时运算过程和最终的降阶系统中也将出现稠密矩阵。其在进行正交化过程时不仅会产生$O(q^2m^2n)$的额外运算，而且会使降阶系统稠密，同时失去结构特性。

单点 SIP 算法

从概念的角度来看，该算法依然是标准的投影方法。但是可以在投影法的结构下实现稀疏矩阵的运算同时达到距匹配的目标。注意两个概念，距匹配的投影方法和计算 Schur 补之间的等价，以及隐式计算稀疏矩阵的 Schur 补。

还用上面的系统为例，矩阵$\boldsymbol{G}$，$\boldsymbol{C}$，$\boldsymbol{E}$是稀疏的，$\boldsymbol{G}$，$\boldsymbol{C}$是对称的。考虑构造一个投影矩阵去匹配 DC（s=0）。令$\boldsymbol{H}(0)$、$\boldsymbol{H}(1)$分别表示原始系统的 order-zero 和 order-one，$\widehat{\boldsymbol{H}}(0)$及$\widehat{\boldsymbol{H}}(1)$表示降阶系统的 order-zero 和 order-one。

引理 2.1

如果$colsp(\boldsymbol{M}) \supset \boldsymbol{V}$,其中 $\boldsymbol{G}\boldsymbol{V} = \boldsymbol{E}$,那么 $\widehat{\boldsymbol{H}}^{(0)}{}_{(0)} = \boldsymbol{H}^{(0)}{}_{(0)}, \widehat{\boldsymbol{H}}^{(1)}{}_{(0)} = \boldsymbol{H}^{(1)}{}_{(0)}$

接下来分割矩阵，分离端口节点

$$\boldsymbol{GV} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1 \\ \boldsymbol{V}_2 \end{bmatrix} = \boldsymbol{E} = \begin{bmatrix} 0 \\ \boldsymbol{I} \end{bmatrix}$$

其中$\boldsymbol{V}_1$表示内部节点的电压，$\boldsymbol{V}_2$表示端口的电压。进一步可得到

$$\boldsymbol{SV}_2 = (\boldsymbol{D} - \boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{B})\boldsymbol{V}_2 = \boldsymbol{I}$$

矩阵$\boldsymbol{S}$为 Schur 补。通过上式得到$\boldsymbol{V}_2$，然后通过下面的引理得到$\boldsymbol{V}_1$

引理 2.2

矩阵$\boldsymbol{M}$定义为

$$\boldsymbol{M} = \begin{bmatrix} -\boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{I} \end{bmatrix}$$

那么$\boldsymbol{V}$可以用下式计算得到

$$\boldsymbol{V} = \boldsymbol{M}\boldsymbol{S}^{-1}$$

引理中的矩阵$\boldsymbol{M}$事实上就是所求的投影矩阵。

直接用 Schur 补的定义来计算矩阵乘法效率很低，特别是对大规模稀疏矩阵来说。下面给出更有效的方法。

定理 2.2

如果$\boldsymbol{G}$ 的 Cholesky 分解为

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_1 & \\ \boldsymbol{L}_3 & \boldsymbol{L}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{L}_1^T & \boldsymbol{L}_3^T \\ & \boldsymbol{L}_2^T \end{bmatrix}$$

那么

$$\boldsymbol{S} = \boldsymbol{L}_2\boldsymbol{L}_2^T$$

由于$\boldsymbol{L}_2$的规模远小于原系统，计算上一步的代价很小，主要的代价在于进行 Cholesky 分解。可以不用先进行 Cholesky 分解再求$\boldsymbol{L}_2\boldsymbol{L}_2^T$，下面是 SIP 算法的关键。

定理 2.3

Schur 补可以写为下面的形式

$$\boldsymbol{S} = \boldsymbol{M}^T\boldsymbol{G}\boldsymbol{M}$$

矩阵$\boldsymbol{M}$的定义见（2. 22）。

定理 2.4

使用（2. 22）定义的矩阵$\boldsymbol{M}$作为投影矩阵, 式（2. 17）能够做到一阶矩匹配。

$$\widehat{\boldsymbol{H}}^{(0)}{}_{(0)} = H^{(0)}{}_{(0)} \ , \ \widehat{\boldsymbol{H}}^{(1)}{}_{(0)} = \boldsymbol{H}^{(1)}{}_{(0)}$$

现在已经展示了 SIP 算法如何计算$\widehat{\boldsymbol{G}}$，$\widehat{\boldsymbol{G}} = \boldsymbol{S}$能够直接从$\boldsymbol{G}$的 Cholesky 分解得到。$\boldsymbol{C}$的降阶矩阵$\widehat{\boldsymbol{C}}$需要更多的计算，将在下面进行介绍。

在一般的投影法中，正交化是确保投影矩阵列满秩的重要步骤。如果投影矩阵的列向量是线性相关的，那么这个模型将是奇异的。SIP 算法中为了避免破坏模型的稀疏性，在执行过程中没有正交化这一步骤，列满秩可以通过下面的定理来保证。

定理 2.5

$\boldsymbol{M}$矩阵是列满秩的。

证明：因为已知

$$\boldsymbol{M}^T\boldsymbol{M} = \boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{I}$$

因为$\boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{A}^{-1}\boldsymbol{B}$可以保证是非负的，$\boldsymbol{I}$是正定的，那么它们的和也是正定的，因此，$\boldsymbol{M}$永远是列满秩的。

现在稀疏矩阵求解的方法已经非常发达，可以有效的减少求解过程中的填入元及预测非零元的位置分布。更重要的是，我们不需要构造显式的$\boldsymbol{M}$，而$\boldsymbol{M}$一般说来是要比$\boldsymbol{L}$稠密的多的，这样就降低了存储空间和 CPU 运行时间。将这两种技术结合起来的方法要比 PRIMA 显式的处理稠密问题要高效的多。

不显式的构造矩阵$\boldsymbol{M}$从而处理矩阵$\boldsymbol{C}$和$\boldsymbol{E}$的关键在于假设$\boldsymbol{G}$和$\boldsymbol{C}$具有相同的非零元分布，如果不是，那么将两者的非零元分布结合，一同做 Cholesky 分解。

标准 Cholesky 分解是递归进行的，令$\boldsymbol{G}(s) = \boldsymbol{G}$，每一步都消去一个变量，当分解到第 i 步时，矩阵$\boldsymbol{G}(i)$具有下面的形式。

$$\boldsymbol{G}^{(i)} = \begin{bmatrix} \boldsymbol{I_{i-1}} & & \\ & a_{i,i} & b_i \\ & \boldsymbol{b}_i^T & B^{(i)} \end{bmatrix}$$

其中$\boldsymbol{I}_{i-1}$是 i-1 维的单位矩阵。要消去节点 i，定义矩阵$\boldsymbol{L}^{(i)}$如下

$$\boldsymbol{L}^{(i)} = \begin{bmatrix} \boldsymbol{I} & & \\ & \sqrt{a_{i,i}} & \\ & -\dfrac{\mathbf{1}}{a_{i,i}}\boldsymbol{b_i} & I_{n-i} \end{bmatrix}$$

那么

$$\boldsymbol{G}^{(i)} = \boldsymbol{L}^{(i)}\boldsymbol{G}^{(i+1)}(\boldsymbol{L}^{(i)})^T$$

$$\boldsymbol{G}^{(i)}=\begin{bmatrix}\boldsymbol{I_{i-1}} & & \\ & 1 & 0 \\ & \mathbf{0} & B^{(i)}-\dfrac{\mathbf{1}}{a_{i,i}}\boldsymbol{b_i}\boldsymbol{b_i}^T\end{bmatrix}$$

重复以上步骤，第n步后$\boldsymbol{G}\ (n+1)\ =\ \boldsymbol{I}$。

那么$\boldsymbol{G}^{(i+1)}\ =\ (\boldsymbol{M}^{(i)})^T\ \boldsymbol{G}^{(i)}\ \boldsymbol{M}^{(i)}$，其中$\boldsymbol{M}^{(i)}=(\boldsymbol{L}^{(i)})^{-T}$。

从上面可以看出，Cholesky 分解可以将内部节点消去，将端口保留。这个结果就是 Schur 补，所以求解 Schur 补不需要将 Cholesky 分解进行完毕，最终的投影实际上是一系列投影的乘积。

$$\boldsymbol{M}=\boldsymbol{M}^{(1)}\boldsymbol{M}^{(2)}\dots\boldsymbol{M}^{(n-m)}$$

这表明可以用更简单的方法对$\boldsymbol{C}$进行降阶投影，只需要递归的对$\boldsymbol{C}$做$\boldsymbol{C}^{(i+1)}\ =\ (\boldsymbol{M}^{(i)})^T\ \boldsymbol{C}^{(i)}\ \boldsymbol{M}^{(i)}$同样的方法即可。

算法 2.4 单点 SIP 算法

Get the optimal ordering to reduce the fill-in of matrix G
move port nodes to the bottom
permute matrices G and C
for $i=1$ to $n-m$ **do**
  Construct matrix $M^{(i)}$
  $G^{(i+1)} \leftarrow {M^{(i)}}^T G^{(i)} M^{(i)}$
  $C^{(i+1)} \leftarrow {M^{(i)}}^T C^{(i)} M^{(i)}$
end for
$\hat{G}=G^{(n-m+1)}(n-m+1:n, n-m+1:n)$
$\hat{C}=C^{(n-m+1)}(n-m+1:n, n-m+1:n)$

多点 SIP 算法

多点 SIP 算法的基础是在不同的频率点做投影，然后将这些投影结合起来。对于一个给定的模型，多点法能够达到单点法的精度，同时因为能够更好的在感兴趣的频带内分布误差而提高效率。同样的，多点法和以前方法之间最大的区别在于能够有效的利用稀疏系统。

对于选定的 q 各频率点$\{s_1\ ,s_2,\cdots,s_q\ \}$,可以定义系统$\boldsymbol{S}=\{\boldsymbol{G}_{\ i}\ ,\boldsymbol{C},\boldsymbol{E}\}$，其中$\boldsymbol{G}_{\ i}=\ \boldsymbol{G}\ +\ \boldsymbol{s}_{\ i}\ \boldsymbol{C}$.这个系统可以写为

$$\begin{cases}(\boldsymbol{G}_{\ i}+\ s\boldsymbol{C})x\ =\ \boldsymbol{E}u \\ \quad y\ =\ \boldsymbol{E}^T\ \ x\end{cases}$$

如果降阶模型匹配第i个变化系统，也就是相当于匹配原系统的$s = s_i$，$\widehat{\boldsymbol{H}}^{(0)}(s_i) = \boldsymbol{H}^{(0)}(s_i)$，$\widehat{\boldsymbol{H}}^{(1)}(s_i) = \boldsymbol{H}^{(1)}(s_i)$。

为了达到目的，可以做 SIP 降阶，如下式

$$\widehat{\boldsymbol{G}}_i = \boldsymbol{M}_i^T \boldsymbol{G}_i \boldsymbol{M}_i,\ \widehat{\boldsymbol{C}}_i = \boldsymbol{M}_i^T \boldsymbol{C}_i \boldsymbol{M}_i$$

在该过程中，下面的投影将隐式的构造

$$\boldsymbol{M}_i = \begin{bmatrix} -\boldsymbol{A}_i^{-1}\boldsymbol{B}_i \\ I \end{bmatrix}$$

其中$\boldsymbol{A}_i = \boldsymbol{A} + s_i\boldsymbol{C}_{11}$，$\boldsymbol{B}_i = \boldsymbol{B} + s_i\boldsymbol{C}_{12}$，所有$\boldsymbol{M}_i$的整合就是多点投影

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{M}_2 & \cdots & \boldsymbol{M}_q \end{bmatrix}$$

最终降阶模型为

$$\widehat{\boldsymbol{G}} = \boldsymbol{M}^T\boldsymbol{G}\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1^T\boldsymbol{G}\boldsymbol{M}_1 & \cdots & \boldsymbol{M}_1^T\boldsymbol{G}\boldsymbol{M}_q \\ \vdots & \ddots & \vdots \\ \boldsymbol{M}_q^T\boldsymbol{G}\boldsymbol{M}_1 & \cdots & \boldsymbol{M}_q^T\boldsymbol{G}\boldsymbol{M}_q \end{bmatrix}$$

$$\widehat{\boldsymbol{C}} = \boldsymbol{M}^T\boldsymbol{C}\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1^T\boldsymbol{C}\boldsymbol{M}_1 & \cdots & \boldsymbol{M}_1^T\boldsymbol{C}\boldsymbol{M}_q \\ \vdots & \ddots & \vdots \\ \boldsymbol{M}_q^T\boldsymbol{C}\boldsymbol{M}_1 & \cdots & \boldsymbol{M}_q^T\boldsymbol{C}\boldsymbol{M}_q \end{bmatrix}$$

定理 2.6

对于任意的 1<i,j<n,始终满足

$$\widehat{\boldsymbol{G}}_i = \boldsymbol{M}_j^T\boldsymbol{G}_i\boldsymbol{M}_i = \boldsymbol{M}_i^T\boldsymbol{G}_i\boldsymbol{M}_j$$

证明

首先证明第一个等式

$$\boldsymbol{M}_j^T\boldsymbol{G}_i\boldsymbol{M}_i = \begin{bmatrix} -\boldsymbol{B}_j^T\boldsymbol{A}_j^{-1} & I \end{bmatrix}\begin{bmatrix} \boldsymbol{A}_i & \boldsymbol{B}_i \\ \boldsymbol{B}_i^T & \boldsymbol{D}_i \end{bmatrix}\begin{bmatrix} -\boldsymbol{B}_i\boldsymbol{A}_i^{-1} \\ I \end{bmatrix}$$

$$= \boldsymbol{D}_i - \boldsymbol{B}_i^T\boldsymbol{A}_i^{-1}\boldsymbol{B}_i = \widehat{\boldsymbol{G}}_i$$

第二个等式很容易证明，这里不做讲解。

因此

$$\widehat{\boldsymbol{G}}_i = \boldsymbol{M}_j^T(\boldsymbol{G} + s_i\boldsymbol{C})\boldsymbol{M}_i$$
$$\widehat{\boldsymbol{G}}_j = \boldsymbol{M}_j^T(\boldsymbol{G} + s_j\boldsymbol{C})\boldsymbol{M}_i$$

所以

$$\widehat{\boldsymbol{C}}_{ji} = \boldsymbol{M}_j^T\boldsymbol{C}\boldsymbol{M}_i = \frac{1}{s_j - s_i}(\widehat{\boldsymbol{G}}_j - \widehat{\boldsymbol{G}}_i)$$

其中$\widehat{\boldsymbol{C}}_{ji}$是矩阵$\widehat{\boldsymbol{C}}$的第（j,i）子矩阵。将$\widehat{\boldsymbol{C}}_{ji}$代入可得

$$\widehat{\boldsymbol{G}}_{ji} = \boldsymbol{M}_i^T \boldsymbol{G}\boldsymbol{M}_j = \widehat{\boldsymbol{G}}_i - s_i\widehat{\boldsymbol{C}}_{ji}$$

易得

$$\widehat{\boldsymbol{C}}_{ji} = \widehat{\boldsymbol{C}}_{ij} \qquad\qquad \widehat{\boldsymbol{G}}_{ji} = \widehat{\boldsymbol{G}}_{ij}$$

算法 2.5 多点 SIP 算法

for $i = 1$ to q **do**
  $[\hat{G}_i,\hat{C}_i]$=SIPcore$(G + s_iC, C,$ ports$)$
end for
for $i = 1$ to q **do**
  for $j = 1$ to i **do**
    $\hat{C}_{ji} = \frac{1}{s_j - s_i}(\hat{G}_j - \hat{G}_i)$
    $\hat{G}_{ji} = \hat{G}_i - s_i\hat{C}_{ji}$
  end for
end for

2.2.2 Pact算法

一个具有 m+1 个端口节点和 n 个内部节点的 RC 网络，可以表示为如下形式

$$(\boldsymbol{G} + s\boldsymbol{C})\boldsymbol{x} = \boldsymbol{b}$$

其中$\boldsymbol{G}$、$\boldsymbol{C}$分别表示电阻和电容矩阵，网络具有 m 个端口，因为端口节点汇总有一个是普通节点。节点电压为$\boldsymbol{x}$，输入电流为$\boldsymbol{b}$。如果电阻和电容是正的，那么$\boldsymbol{G}$、$\boldsymbol{C}$是对角占优矩阵。这个条件并不能保证矩阵是对称正定的，对称正定阵要求矩阵所有的特征值为正。是否对称正定关系到系统能否保持无源性。

对该多输入端口系统进行排序，将 m 个端口节点放置于前，n 个内部节点放置于后。可以写作

$$\left(\begin{bmatrix}\boldsymbol{A} & \boldsymbol{Q}^T\\ \boldsymbol{Q} & \boldsymbol{D}\end{bmatrix} + s\begin{bmatrix}\boldsymbol{B} & \boldsymbol{R}^T\\ \boldsymbol{R} & \boldsymbol{E}\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{x}'\\ \boldsymbol{x}''\end{bmatrix} = \begin{bmatrix}\boldsymbol{b}'\\ 0\end{bmatrix}$$

$\boldsymbol{x}'$和$\boldsymbol{x}''$分别表示端口节点与内部节点的电压。$\boldsymbol{b}$中表示内部节点的值为 0，因为输入不会到达这些节点。$m \times m$维的矩阵$\boldsymbol{A}$和$\boldsymbol{B}$是端口节点矩阵，描述了端口之间的相互关系。$n \times n$维的矩阵$\boldsymbol{D}$和$\boldsymbol{E}$是内部矩阵。$n \times m$维的矩阵$\boldsymbol{Q}$和$\boldsymbol{R}$描述

了端口节点和内部节点之间的相互关系，称为连接矩阵。矩阵$\boldsymbol{A}$、$\boldsymbol{B}$和$\boldsymbol{E}$是对称半正定的，$\boldsymbol{D}$是对称正定的。

利用定义$\boldsymbol{Y}(\mathrm{s})\boldsymbol{x}^{'}=\boldsymbol{b}^{'}$，消去$\boldsymbol{x}^{''}$得到

$$\boldsymbol{Y}(s)=\boldsymbol{A}+s\boldsymbol{B}-(\boldsymbol{Q}+s\boldsymbol{R})^{T}(\boldsymbol{D}+s\boldsymbol{E})^{-1}(\boldsymbol{Q}+s\boldsymbol{R})$$

当$(\boldsymbol{D}+s\boldsymbol{E})$为奇异时，$\boldsymbol{Y}(s)$出现极点，相当于解下面的式子得到的特征值$\lambda$，然后$-\lambda-1$即为所需的值。

$$det[\boldsymbol{E}-\lambda\boldsymbol{D}]=0$$

因为$\boldsymbol{E}$是对称半正定的，而$\boldsymbol{D}$是对称正定的，那么$\boldsymbol{Y}(s)$的极点是负实数。

方阵$\boldsymbol{W}$的全等变换可以表示为$\boldsymbol{X}=\boldsymbol{V}^{T}\boldsymbol{W}\boldsymbol{V}$。经过全等变换后，矩阵的特征值不变，举例来说，选择合适的$\boldsymbol{V}$对矩阵$\boldsymbol{E}$、$\boldsymbol{D}$做全等变换，下式求得的特征值与（2. 44）相同。

$$det[\boldsymbol{V}^{T}\boldsymbol{E}\boldsymbol{V}-\lambda\boldsymbol{V}^{T}\boldsymbol{D}\boldsymbol{V}]=0$$

那么极点也是相同的。这样的话 RC 网络的电阻和电容矩阵$\boldsymbol{G}$、$\boldsymbol{C}$就可以通过全等变换进行降阶，将维度消减下去。同时任何具有无源性的 RC 网络经过全等变换依然可以保持其无源性。

首先，用全等变换转换矩阵$\boldsymbol{D}$，消去连接矩阵$\boldsymbol{Q}$，这是通过 Cholesky 分解，$\boldsymbol{L}\boldsymbol{L}^{T}=\boldsymbol{D}$实现的（$\boldsymbol{L}$是下三角阵）。

$$\boldsymbol{V}=\begin{bmatrix}\boldsymbol{I} & 0\\ -\boldsymbol{X} & \boldsymbol{L}^{-T}\end{bmatrix}$$

网络对应的转换是

$$\boldsymbol{G}'=\boldsymbol{V}^{T}\boldsymbol{G}\boldsymbol{V}=\begin{bmatrix}\boldsymbol{A}-\boldsymbol{Q}^{T}\boldsymbol{X} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{L^{-1}D}\boldsymbol{L^{-T}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}' & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}$$

$$\boldsymbol{C}'=\boldsymbol{V}^{T}\boldsymbol{C}\boldsymbol{V}=\begin{bmatrix}\boldsymbol{B}-\boldsymbol{P}^{T}\boldsymbol{X}-\boldsymbol{X}^{T}\boldsymbol{R} & \boldsymbol{P}^{T}\boldsymbol{L}^{-T}\\ \boldsymbol{L^{-1}P} & \boldsymbol{L^{-1}E}\boldsymbol{L^{-T}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{B}' & \boldsymbol{R}'^{\boldsymbol{T}}\\ \boldsymbol{R}' & \boldsymbol{E}'\end{bmatrix}$$

其中$\boldsymbol{X}=\boldsymbol{D^{-1}Q}$，$\boldsymbol{W}=\boldsymbol{E}\boldsymbol{X}$，$\boldsymbol{P}=\boldsymbol{R}-\boldsymbol{W}$是中间变量。那么，方程（2. 43）可以使用$\boldsymbol{G}'$和$\boldsymbol{C}'$的部分写作

$$\boldsymbol{Y}(s)=\boldsymbol{A}'+s\boldsymbol{B}'-s^{2}\boldsymbol{R}'^{\boldsymbol{T}}(\boldsymbol{I}+s\boldsymbol{E}')^{-1}\boldsymbol{R}'$$

因为网络的极点出现在$(\boldsymbol{I}+s\boldsymbol{E}')$时，那么可以通过将$\boldsymbol{E}'$对角化使之分离，这个过程可以通过特征分解$\boldsymbol{E}'=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{T}$实现。对角阵$\boldsymbol{\Lambda}$包含了$\boldsymbol{E}'$的特征值，$\boldsymbol{U}$是由对应的特征向量所构成的方阵，并且是正交化的。

$$\boldsymbol{G}''=\begin{bmatrix}\boldsymbol{I} & 0\\ 0 & \boldsymbol{U}^{T}\end{bmatrix}\begin{bmatrix}\boldsymbol{A}' & 0\\ 0 & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & 0\\ 0 & \boldsymbol{U}\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}' & 0\\ 0 & \boldsymbol{I}\end{bmatrix}$$

$$\boldsymbol{C''} = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{U}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{B'} & \boldsymbol{R'}^{\boldsymbol{T}} \\ \boldsymbol{R'} & \boldsymbol{E'} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{U} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B'} & \boldsymbol{R''}^{\boldsymbol{T}} \\ \boldsymbol{R''} & \boldsymbol{E''} \end{bmatrix}$$

所以$\boldsymbol{E''} = \boldsymbol{U}^T \boldsymbol{E'} \boldsymbol{U} = \boldsymbol{\Lambda}$，其中$\boldsymbol{\Lambda}$是包含了$\boldsymbol{E'}$的特征值的对角阵。因为$\boldsymbol{U}$是非奇异的方阵，所以$\boldsymbol{Y}(s)$不变。现在两个内部矩阵都对角化了，（2. 49）可以写作

$$\mathrm{Y(s)} = \boldsymbol{A'} + \mathrm{s}\boldsymbol{B'} - \frac{s^2 \boldsymbol{r}_1^T \boldsymbol{r}_1}{1 + se_{11}} - \cdots - \frac{s^2 \boldsymbol{r}_n^T \boldsymbol{r}_n}{1 + se_{nn}}$$

其中$\boldsymbol{r}_i$是$\boldsymbol{R''}$的第 i 行，e_{ii}是$\boldsymbol{E''}$的第 i 个对角元。这样就可以在不对整个系统产生显著影响的情况下将极点频率大于设定值$2\pi f_c = {\lambda_c}^{-1}$的频点消去，达到降低问题规模的目的。具体就是将$\boldsymbol{G''}$和$\boldsymbol{C''}$中对应$\boldsymbol{E''}$小于$\lambda_c$的对角元位置的行与列消去。

Lanczos 算法是寻找大规模对称矩阵最大特征值及对应的特征向量的有效办法。在进行（2. 50）（2. 51）的变换之前就可以将$\boldsymbol{U}$中对应不需要的极点的行、列消去，这样就只剩下一个小规模的$\boldsymbol{E'}$的特征值大于λ_c子集需要处理。其中一个重要的性质就是不需要对$\boldsymbol{E'}$进行修正，这样问题的稀疏性可以得到保持，$\boldsymbol{E'x}$通过$\boldsymbol{L^{-1}} \boldsymbol{E} \boldsymbol{L^{-T}} \boldsymbol{x}$来计算。

基本的 Lanczos 算法通过以下递归实现

$$\widetilde{\boldsymbol{w}}_{j+1} = A\boldsymbol{w}_j - \alpha_j \boldsymbol{w}_j - \beta_{j-1} \boldsymbol{w}_{j-1}$$

其中

$$\alpha_j = {\boldsymbol{w}_j}^T \boldsymbol{A} \boldsymbol{w}_j$$

$$\beta_j = \left\| \widetilde{\boldsymbol{w}}_{j+1} \right\|_2$$

那么

$$\boldsymbol{w}_{j+1} = \frac{\widetilde{\boldsymbol{w}}_{j+1}}{\beta_j}$$

递归开始时，$\boldsymbol{w}_1$可以任意选择模为 1 的向量，β设为 0。待处理的矩阵是$\boldsymbol{A}$，向量$\boldsymbol{w}_1 \cdots \boldsymbol{w}_k$称为 Lanczos 向量。这些向量是正交的，也就是说${\boldsymbol{w}_i}^T \boldsymbol{w}_j = 0$，当$i \neq j$时，${\boldsymbol{w}_i}^T \boldsymbol{w}_j = 1$，当$i = j$时。同一时间运算存储中只需要保留两个向量。

$$\boldsymbol{T} = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix}$$

$\boldsymbol{T}$的特征值近似与$\boldsymbol{A}$，称为 Ritz 值。这个值一般先收敛到$\boldsymbol{A}$的最大特征值，

然后计算其他的向量直到该值收敛到一定的范围。$\boldsymbol{A}$的近似特征向量称为 Ritz 向量，可以通过下式得到

$$\boldsymbol{U} = \boldsymbol{W}\boldsymbol{Z}$$

其中$\boldsymbol{Z}$是$\boldsymbol{T}$的$k \times k$的特征向量矩阵，$\boldsymbol{W}$中的 k 列是 Lanczos 向量，$\boldsymbol{U}$中的列是对应第 k 个 Ritz 值的 Ritz 特征向量。该算法有两个缺点，一是只能找到一组特征值中的一个，二是当有 2 个或更多个相距较近的特征值时收敛较慢。

2.2.3 TICER算法

TICER（time constant equilibration reduction）算法[52]也是适用于大规模 RC 网络模型降阶的算法，它的核心思想是根据实际情况的需要，以节点的时间常数为标准，将节点分类，然后将不符合要求的节点消去。

考虑一个星型的网络，由 1 个中心节点和 N 个（编号为 0 到 N-1）端口组成。用g_{iN}和c_{iN}分别表示第 i 个端口与中心节点间的电感与电容。

当给第 i 个端口施加激励电压时，中心节点的响应见下式，其余的端口相当于接地。

$$h_{iN}(t) = \frac{g_{iN}}{\gamma_N} + (\frac{c_{iN}}{\chi_N} - \frac{g_{iN}}{\gamma_N})e^{-t/\tau_N}$$

其中

$$\gamma_N = \sum_{k=0}^{N-1} g_{kN}\,, \quad \chi_N = \sum_{k=0}^{N-1} c_{kN} \quad , \quad \tau_N = \frac{\chi_N}{\gamma_N}$$

通过式 （2. 59）可以得到中心节点对任意端口的响应，相应的可以得到中心节点的特征时间常数。

$$\tau_N = \sum_{k=0}^{N-1} c_{kN} \Big/ \sum_{k=0}^{N-1} g_{kN}$$

这个时间常数是独立于节点邻居及邻居的组合的。一个节点的时间常数就是该节点对其他节点及对地的总电容除以该节点对其他节点及对地的总电导。

现在对于一个任意的 RC 网络，在保持一定其一定频率范围内特性的前提下进行降阶。将节点根据其时间常数分为小于、大于或介于所规定的频率范围三种情况。称为快速、慢速或一般节点。

这样分类的原因在于，消去快速及慢速节点的话，对于系统性能没有明显的影响，起码在所规定的频率范围内影响不大。原电路的时间常数可能跨越很

宽的动态范围，而在平衡电路中，它将集中在指定的频域附近。

接下来消去快速及慢速节点，首先对 RC 电路进行拉普拉斯变换：

$$(\boldsymbol{C}s+\boldsymbol{G})\boldsymbol{v}=\boldsymbol{Y}\boldsymbol{v}=\boldsymbol{J}$$

其中$\boldsymbol{C}\in\boldsymbol{R}^{N\times N}$，$\boldsymbol{G}\in\boldsymbol{R}^{N\times N}$分别是节点的电容和电导矩阵，$\boldsymbol{v}\in\boldsymbol{R}^{N}$是节点的电压向量，$\boldsymbol{J}\in\boldsymbol{R}^{N}$来自节点的输入。

简单来说，假设想要消去的节点是 N，将上式写作分块系统得

$$\begin{bmatrix}\widetilde{\boldsymbol{Y}} & \boldsymbol{y}\\ \boldsymbol{y}^{T} & (\gamma_N+s\ \chi_N)\end{bmatrix}\begin{bmatrix}\widetilde{\boldsymbol{v}}\\ \boldsymbol{v}_N\end{bmatrix}=\begin{bmatrix}\tilde{\boldsymbol{J}}\\ \boldsymbol{j}_N\end{bmatrix}$$

从第二个块方程解出$\boldsymbol{v}_N$代入第一个块方程中得到

$$\left(\widetilde{\boldsymbol{Y}}-\boldsymbol{E}\right)\boldsymbol{v}_N=\tilde{\boldsymbol{J}}-\boldsymbol{F}$$

$$\boldsymbol{E}_{ij}=\frac{(g_{iN}+sc_{iN})(g_{jN}+sc_{jN})}{\gamma_N+s\ \chi_N}$$

$$\boldsymbol{F}_i=\frac{(g_{iN}+sc_{iN})}{\gamma_N+s\ \chi_N}j_N$$

如果有的情况下γ_N、χ_N为零，那么j_N和$\boldsymbol{F}$就为零。

当$s\ \chi_N\ll\gamma_N$时，将要消去的是快速节点，那么近似$\boldsymbol{E}_{ij}$见下式

$$\boldsymbol{E}_{ij}\approx\frac{g_{iN}g_{jN}}{\gamma_N}\left(1-\frac{s\ \chi_N}{\gamma_N}\right)+s\frac{g_{iN}g_{jN}+g_{jN}c_{iN}}{\gamma_N}$$

上式可以看作是对电路进行物理上的修改。要消去一个网络中的快速节点 N，先将与之相连的其它节点的电阻和电容移除，然后在节点 N 从前的邻居之间增加电阻和电容。如果节点 i 与节点 j 同节点 N 之间有电导g_{iN}与g_{jN}，那么插入一个电导$g_{iN}g_{jN}/\gamma_N$至节点 i 与 j 之间；如果节点 i 与 N 之间有电容c_{iN}，节点 j 与 N 中间有电导g_{jN}，那么在节点 i 与 j 之间插入一个电容$c_{iN}g_{jN}/\gamma_N$。

节点 N 所有时刻的电压可以写作

$$\mathbf{v}_N(t)=\frac{1}{\gamma_N}\sum_{j=0}^{N-1}g_{jN}\,\mathbf{v}_j(t)$$

如果保持这个状态，那么从节点 i 经由g_{iN}进入节点 N 的电流为

$$i_{iN}(t)=\sum_{j=0}^{N-1}\frac{g_{iN}g_{jN}}{\gamma_N}[\boldsymbol{v}_i(t)-\boldsymbol{v}_j(t)]$$

网络的在电容c_{iN}上的负载相当于

$$q_{iN}(t)=\sum_{j=0}^{N-1}\frac{c_{iN}g_{jN}}{\gamma_N}[v_i(t)-v_j(t)]$$

当$s\ \chi_N \gg \gamma_N$时，消去的节点是慢速节点，那么近似E_{ij}见下式

$$\boldsymbol{E}_{ij} \approx \frac{g_{iN}g_{jN}}{\gamma_N} + s\frac{c_{iN}c_{jN}}{\gamma_N}$$

这个方程是将下式代入　　（2. 64）得到的

$$\frac{1}{\gamma_N + s\ \chi_N} \approx \frac{1}{s\ \chi_N}\left(1 - \frac{\gamma_N}{s\ \chi_N}\right)$$

要消去慢速节点 N，首先移除连接节点 N 与其他节点的电阻和电容，如果节点 i 与节点 j 同节点 N 之间有电导g_{iN}与g_{jN}，那么插入一个电导$g_{iN}g_{jN}/\gamma_N$至节点 i 与 j 之间；如果节点 i 与 N 之间有电容c_{iN}，节点 j 与 N 中间有电容c_{jN}，那么在节点 i 与 j 之间插入一个电容$c_{iN}c_{jN}/\gamma_N$。

这样逐个消去快速与慢速节点，直到最后得到降阶系统。

第三章 基于优化消去的大规模 RC 网络模型降阶算法

3.1 问题背景

随着集成电路复杂度以及电路规模的提高，电路仿真验证遇到极大挑战。即使对于中等规模的电路，电路后仿（Post Simulation），也就是在寄生参数提取之后的电路仿真，需要求解的电路节点数也轻易达到百万以上的量级。而对典型的混合信号电路进行整体后仿，规模往往超过千万甚至上亿节点。这对仿真器而言是个极大的挑战。

因此，人们希望通过模型降阶（Model Order Reduction）来减小电路的规模，同时又希望由此带来对精度的牺牲尽量小。在数字电路设计中模型降阶已经被广泛采用。标准的模型降阶方法，如 PRIMA[28]，Poor Man's TBR[46]，Rational-Krylov[16]等，可以将几千甚至上万节点的 RC 网络降阶到只有几个元件，同时可以保证电路分析所需的精度。然而同样的方法对寄生参数提取后的模拟与混合信号电路却完全无能为力。其原因在于，这些传统方法只能处理端口（也就是对外连接的节点）数很小的 RC 系统。典型的数字电路中需要关心的互连电路，通常来讲只有很少的几个端口，用于描述互连的输入，输出，串扰等等。对于这样的电路，传统的模型降阶方法具有很好的效果。当电路的端口数增大时，传统降阶方法的弊端就显现出来了。由于传统降阶方法得到的系统往往为稠密系统，当端口数增大时，一方面降阶算法的效率急剧下降，另一方面降阶后的模型复杂度也迅速上升。并且“降阶”后的模型复杂度往往远远大于原始电路的复杂度，虽然其电路节点数小于原始电路。这使得“降阶”变得毫无意义。

近年来，研究者针对上述两个方面的问题，提出了一系列的解决方法[24][49-53]。其中，由于具有能够同时解决以上两个问题的潜力，基于节点消去的方法[24][52][53]一般被认为是针对大规模 RC 系统唯一可能有效的一类方法。现有的基于节点消去的模型降阶方法基本思想是逐个消去网络中的节点。在消去每个节点的同时，对该节点的近邻节点两两之间建立一个连接，连接的 RC 支路的数值根据矩匹配的规则进行确定。基于节点消去的方法，得到的系统往往还具有一定的稀疏性。

虽然基于节点消去的方法相对于传统的降阶方法得到的模型更为稀疏，然而随着节点的消去，一般而言网络仍然会变得更加稠密。盲目消去节点往往会导致“降阶”后的电路比原始电路更加复杂。因此节点数的降低与网络复杂度（可以以矩阵非零元数目作为衡量）之间存在一个矛盾，势必可以在节点数与网络复杂度之间找到某种折中使得降阶后的电路仿真速度达到最高。

下面我们提出一种基于优化消去的模型降阶方法。首先快速得到大规模 RC 网络的剩余节点数与非零元的关系，其次采用统计建模的方法，得到网络求解时间与节点数和非零元的关系。通过这两个关系可以直接得到最优的节点消去策略。实验表明采用该策略可使得降阶网络的求解速度提高 2~5 倍。同时此方法也可以与其他的降阶方法相结合，以降低降阶复杂度。在后续几节中我们依次介绍网络约减技术及其在电路后仿中的应用、基于优化消去的模型降阶方法和实验结果。

3.2 RC 网络约减在电路后仿中的应用

在混合信号电路后仿的过程中，不但要考虑线网的电阻（R）以及自电容（C），还需考虑线网之间的耦合电容（CC），我们称之为“考虑 C+CC+R”的后仿真。图 3.1 显示了一个考虑 C+CC+R 情况的电路的例子，节点数(问题规模)大大增加，不仅增加了寄生电阻，而且寄生电容数目也会极大的增加，以至于大规模的模拟与混合信号电路无法进行直接仿真，所以必须对模型进行降阶处理，对于大型 RC 网络就是网络约减。目的是输入大规模 RC 网络，输出端口特性近似输入网络，且保持电路无源性的小规模 RC 网络。

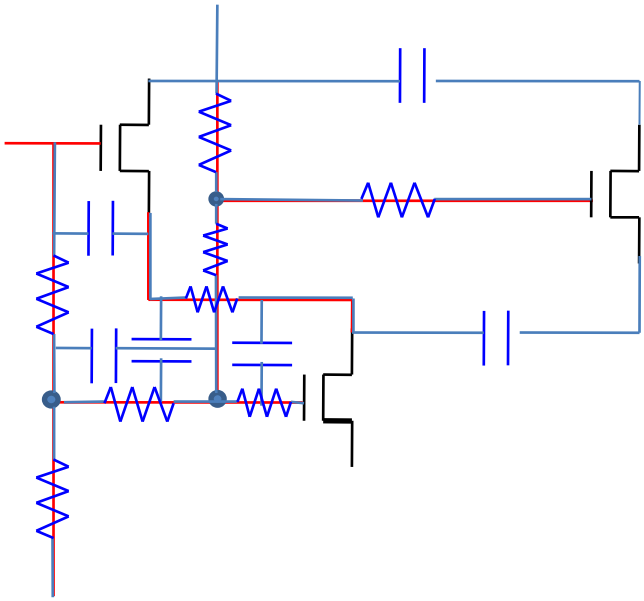


图 3.1.考虑 C+CC+R 情况的 RC 网

3.2.1投影法

将大规模 RC 网络系统经过标准的节点分析，可以写成下面的形式：

$$\begin{cases} \boldsymbol{E}\boldsymbol{u}(t) = \boldsymbol{C}\dot{\boldsymbol{x}}(t) + \boldsymbol{G}\boldsymbol{x}(t) \\ \boldsymbol{y}(t) = \boldsymbol{E}^{T}\boldsymbol{x}(t) \end{cases}$$

其中$\boldsymbol{C}$是 RC 网络系统的电容矩阵，$\boldsymbol{G}$是 RC 网络系统的电阻矩阵。

通过 Krylov 子空间等投影方法可以对系统（3. 1）进行降阶。通过构造不同的子空间，可以得到不同的变换矩阵$\boldsymbol{V}$，$\boldsymbol{W} \in \boldsymbol{E}^{n\times r}$，根据变换矩阵$\boldsymbol{V}$和$\boldsymbol{W}$，就可以得到原始 RC 网络系统的降阶系统（3. 2）为：

$$\begin{cases} \widetilde{\boldsymbol{E}}\tilde{\boldsymbol{u}}(t) = \widetilde{\boldsymbol{C}}\dot{\tilde{\boldsymbol{x}}}(t) + \widetilde{\boldsymbol{G}}\tilde{\boldsymbol{x}}(t) \\ \tilde{\boldsymbol{y}}(t) = \widetilde{\boldsymbol{E}}^{\mathrm{T}}\tilde{\boldsymbol{x}}(t) \end{cases}$$

其中$\tilde{\boldsymbol{x}}(t), \dot{\tilde{\boldsymbol{x}}}(t) \in \boldsymbol{R}^{r}$，$\widetilde{\boldsymbol{E}} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{E}\boldsymbol{V}$，$\widetilde{\boldsymbol{C}} = \boldsymbol{W}^{T}\boldsymbol{C}\boldsymbol{V}$，$\widetilde{\boldsymbol{G}} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{V}$，此系统即为（3. 1）的降阶系统。投影法的优点在于精度高，但是会破坏系统的原有特性，比如矩阵的稀疏性以及形状分布的特点无法得到保持。

3.2.2节点消去法

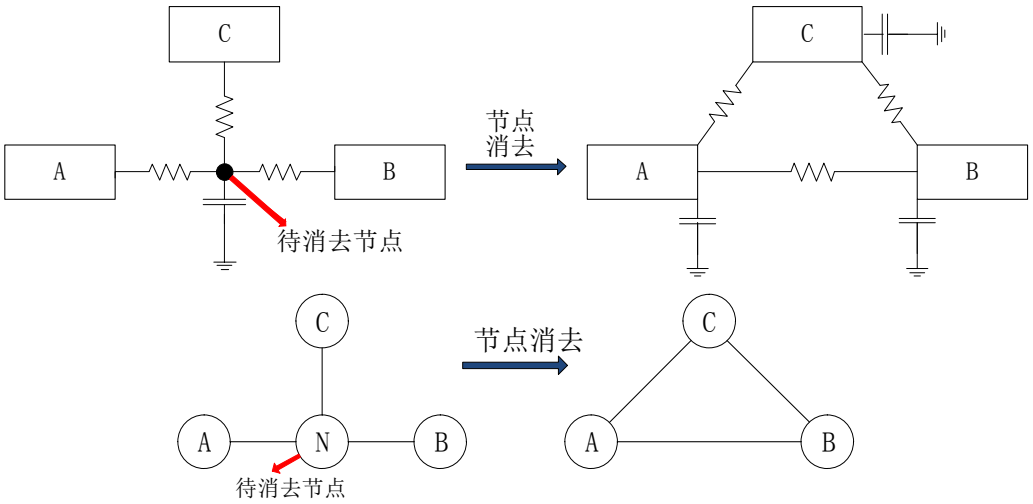


图 3.2.RC 网络节点消去的物理模型和图的表示

对大型 RC 网络进行约减的节点消去法，通过消去部分选定的节点以降低问题的维度。从图 3.2 可以看出，消去一个节点，器件之间的关联需要进行修改，与被消去节点相连的节点要两两连接。当消去节点 N 时，同时消去了边 AN、BN、CN，在新的图中要将节点 A、B、C 连接起来，产生了新的边 AB、AC、BC。

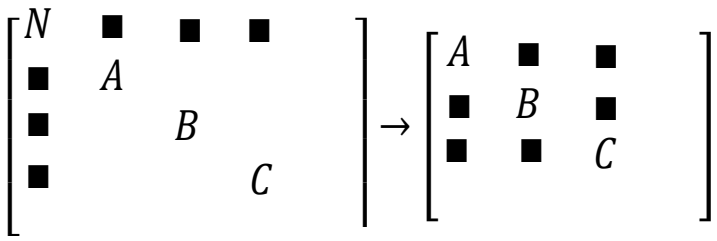


图 3.3RC 网络节点消去的矩阵表示

图 3.3 是用矩阵形式表示节点消去的过程：将各个节点作为矩阵的对角元，其中的 6 个黑色方块表示边 NA、NB、NC。当消去节点 N 时，矩阵变为图 3.3 右侧所示，表示边 NA、NB、NC 的黑色方块随着节点 N 消去，但是新产生了表示边 AB、AC、BC 的黑色方块。使用矩阵同样可以很好的表示电路的拓扑结构，而且可以提高实际约减过程的速度和可操作性。

设$\boldsymbol{G}$、$\boldsymbol{C}$为别为 RC 网络的电阻、电容矩阵，两者皆为对称正定矩阵。对它们进行电路节点消去的过程与对矩阵执行高斯消去法是完全等价的。以$\boldsymbol{G}$为例，

令$\boldsymbol{G}^{(0)}=\boldsymbol{G}$，消去一个节点，矩阵记为$\boldsymbol{G}^{(1)}$，消去 i 个节点，矩阵记为$\boldsymbol{G}^{(i)}$。

$$\boldsymbol{G}^{(0)}=\boldsymbol{G}=\begin{bmatrix}a_{11} & \boldsymbol{b}_0\\ \boldsymbol{b}_0^T & \boldsymbol{B}^{(0)}\end{bmatrix}$$

定义第一个消去阵为：

$$\boldsymbol{L}^{(0)}=\begin{bmatrix}\sqrt{a_{11}} & \\ -\dfrac{1}{a_{1,1}}\boldsymbol{b}_0^T & \boldsymbol{I}_{n-1}\end{bmatrix}$$

那么$\boldsymbol{G}^{(1)}$可以通过（3. 5）式计算得到：

$$\boldsymbol{G}^{(1)}=\boldsymbol{L}^{(0)}\boldsymbol{G}^{(0)}\left(\boldsymbol{L}^{(0)}\right)^{\mathrm{T}}=\begin{bmatrix}1 & 0\\ 0 & \boldsymbol{B}^{(0)}-(-\dfrac{1}{a_{1,1}})\boldsymbol{b}_0\boldsymbol{b}_0^T\end{bmatrix}$$

同理可得到：

$$\boldsymbol{L}^{(i)}=\begin{bmatrix}\boldsymbol{I} & & \\ & \sqrt{a_{11}} & \\ & -\dfrac{1}{a_{i,i}}\boldsymbol{b}_i^T & \boldsymbol{I}_{n-i}\end{bmatrix}$$

$$\boldsymbol{G}^{(i)}=\begin{bmatrix}\boldsymbol{I}_{i-1} & & \\ & a_{i,i} & \boldsymbol{b}_i\\ & \boldsymbol{b}_i^T & \boldsymbol{B}^{(i)}\end{bmatrix}$$

$$\boldsymbol{G}^{(i+1)}=\boldsymbol{L}^{(i)}\boldsymbol{G}^{(i)}\left(\boldsymbol{L}^{(i)}\right)^{T}=\begin{bmatrix}\boldsymbol{I}_{i-1} & & \\ & 1 & 0\\ & 0 & \boldsymbol{B}^{(i)}-\left(-\dfrac{1}{a_{i,i}}\right)\boldsymbol{b}_i\boldsymbol{b}_i^T\end{bmatrix}$$

那么消去 i+1 个节点后的降阶矩阵为：

$$\boldsymbol{B}^{(i)}-\left(-\frac{1}{a_{i,i}}\right)\boldsymbol{b}_i\boldsymbol{b}_i^T$$

从式（3. 7）到式（3. 8）又称为计算一步的 Schur 补（Schur Complement）。持续进行这样的运算可以将矩阵的节点按照其排序的顺序进行消去。现有的节点消去法在一定程度上能够保证模型的稀疏性，但是由于每消去一个节点都要对整个矩阵进行重新计算，运算时间过长，不能很好的满足实际问题的需要。

3.3．基于优化消去的模型降阶方法

3.3.1实际问题的特点和处理的重点

RC 网络对应的矩阵是对称稀疏矩阵，只需要处理矩阵的上三角或下三角部

分，而且非零元很少，处理过程中可以采取具有针对性的方法以提高运算效率，减少运算时间。本方法只需考虑其中元素的数目和分布，不用做实际的数值计算，可以大大的减少运算量。

对 RC 网络进行节点消去，目的是在精度允许的范围内降低模型的阶数，减少电路后仿的时间。RC 矩阵在消减的过程中会产生数量不定的非零元，有可能当节点消减到一定程度后，因为新增连接太多反而使得运算时间大大增加。必须要在 RC 网络保留的节点数和增加的连接数之间做一个评估，确定一个最佳的消去程度，使得消减得到的 RC 网络具有最小的处理时间。

3.3.2快速遍历统计非零元数量

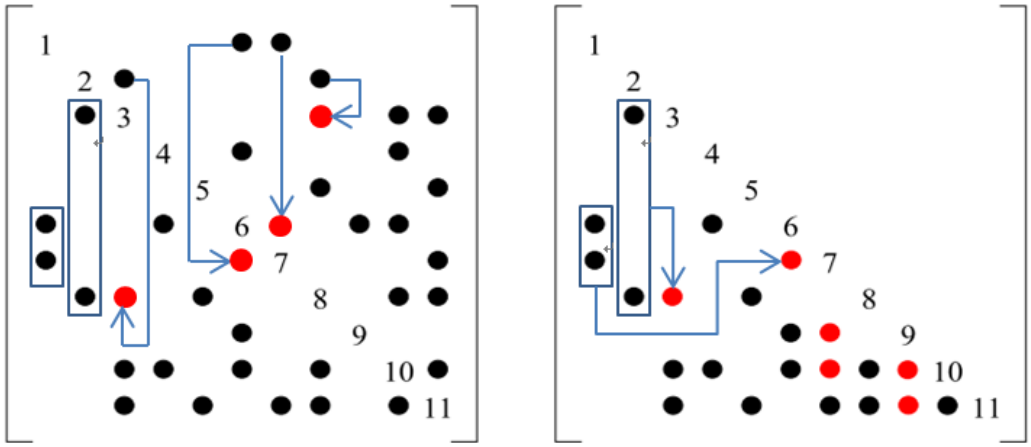


图 3.4. RC 网络矩阵的节点消去过程

图 3.4 说明了节点消去过程中如何确定非零元的位置，红色的点是消去过程中产生的填入元。矩阵$\boldsymbol{A}$是一个对称矩阵，表示一个简单的 RC 网络。当消去第一列的元素$\boldsymbol{A}(6,1)$时，由于元素$\boldsymbol{A}(1,7)$的存在，点$\boldsymbol{A}(6,7)$会产生一个填入元，当消去$\boldsymbol{A}(7,1)$时，由于元素$\boldsymbol{A}(1,6)$的存在，点$\boldsymbol{A}(7,6)$会产生一个填入元。同理点$\boldsymbol{A}(3,8)$、$\boldsymbol{A}(8,3)$是消去第二列元素时产生的填入元。由于矩阵$\boldsymbol{A}$是对称矩阵，产生的填入元位置也是对称的，只需要对矩阵的上三角或下三角部分进行操作即可。见图 4 右侧下三角阵，可以理解为$\boldsymbol{A}(6,1)$与$\boldsymbol{A}(7,1)$产生了填入元$\boldsymbol{A}(7,6)$，$\boldsymbol{A}(3,2)$与$\boldsymbol{A}(8,2)$产生了填入元$\boldsymbol{A}(8,3)$。

那么，当消去第三列时，由于该列非零元位于第 8、10、11 列，那么产生的填入元将位于$\boldsymbol{A}(10,8)$、$\boldsymbol{A}(11,8)$、$\boldsymbol{A}(11,10)$，但是由于这三个位置已有元素，所以不产生填入。消去某列元素时，根据该列元素的行号即可确定填入元的位置，

产生的填入元在消去其所在列时也可能产生新的填入元。

当对大规模的 RC 系统进行处理时，将 RC 系统用矩阵来表示，生成一个对称的稀疏矩阵，使用压缩稀疏行（CSR）的格式存储矩阵（如图 3.5），用$\boldsymbol{Ap}$，$\boldsymbol{Ai}$，$\boldsymbol{Ax}$三个向量来存储维度为 n×n，非零元数目为 nz 的矩阵$\boldsymbol{A}$。$\boldsymbol{Ap}$是长度为 n+1 的向量，$\boldsymbol{Ap}$(i)表示矩阵第 1 至 i-1 行的非零元数目之和，$\boldsymbol{Ap}(1) = 0$。$\boldsymbol{Ai}$及$\boldsymbol{Ax}$是长度为 nz 的向量，$\boldsymbol{Ax}$中存储的是从第一行开始，列号由小至大的非零元的数值，$\boldsymbol{Ai}$(i)中存储的是$\boldsymbol{Ax}$(i)的列号。采用这样的存储方法，可以减少存储空间，矩阵$\boldsymbol{A}$所占用的空间只有 2×nz+n。同时也可以方便我们进行约减，遍历查找时可以减少循环次数，提高运算速度。

$$\boldsymbol{A} = \begin{bmatrix} 4.5 & 0 & 3.2 & 0 \\ 3.1 & 2.9 & 0 & 0.9 \\ 0 & 1.7 & 3.0 & 0 \\ 3.5 & 0.4 & 0 & 1.0 \end{bmatrix}$$

$$\boldsymbol{Ap} = [0, \quad 2, \quad 5, \quad 7, \quad 10]$$

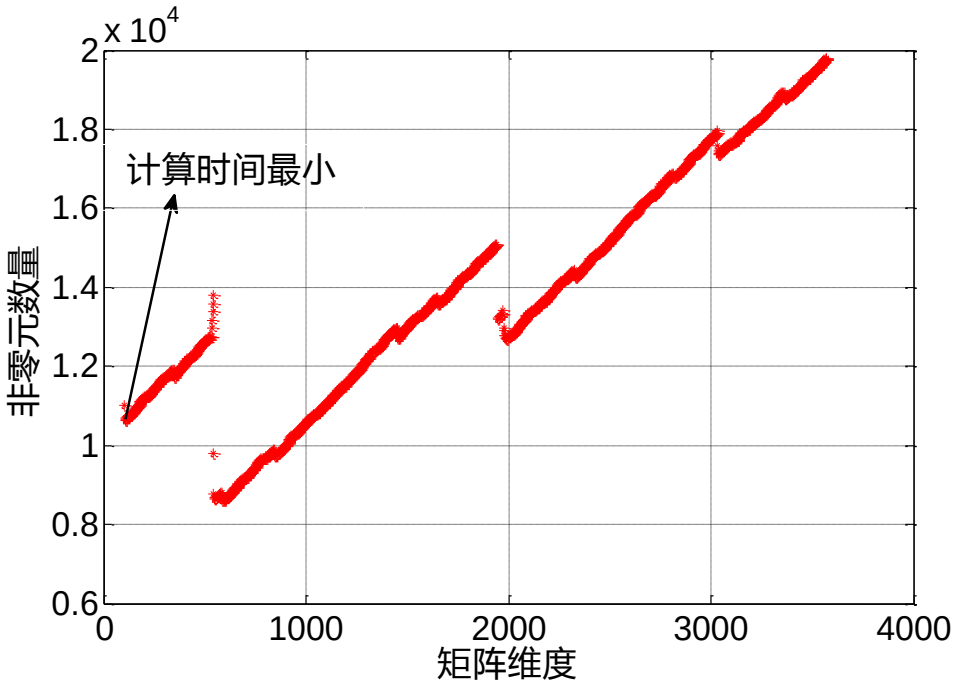
$$\boldsymbol{Ai} = [0, \quad 2, \quad 0, \quad 1, \quad 3, \quad 1, \quad 2, \quad 0, \quad 1, \quad 3]$$

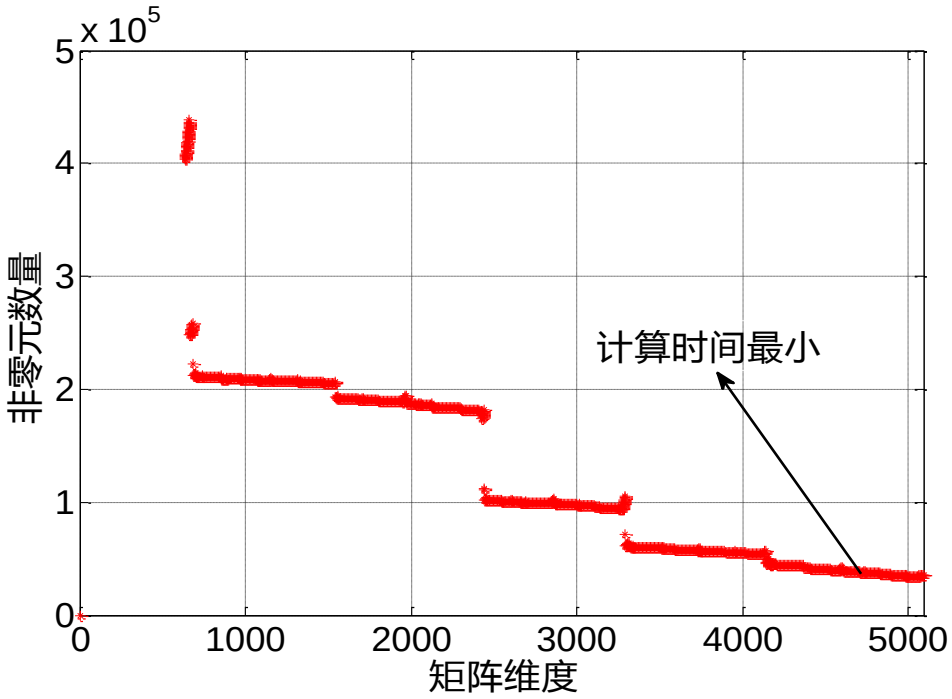
$$\boldsymbol{Ax} = [4.5, \quad 3.2, \quad 3.1, \quad 2.9, \quad 0.9, \quad 1.7, \quad 3.0, \quad 3.5, \quad 0.4, \quad 1.0]$$

图 3.5.矩阵转换为压缩稀疏行形式

按照以上的存储方法，在实际的运算中存在一个问题，在消去元素和产生填入元的过程中，需要不断的改变向量$\boldsymbol{Ai}$、$\boldsymbol{Ax}$的长度来进行消去和插入，这样使得运算的效率很低，应当在建立向量的同时预测在整个过程中可能产生的填入元的数目，提前申请空间，在计算的过程中只要修改对应的数值即可，不用改变向量的长度，从而减少运算时间。可以对 RC 矩阵进行一次 Cholesky 分解，得到的上三角阵$\boldsymbol{U}$即为填入元最多的情况，将$\boldsymbol{U}$用 CSR 格式存储即可。

消去的主要过程是先对矩阵进行一次强制最小度排序（Camd）[54]，再做一次 Cholesky 分解，然后按照 CSR 存储的格式存储$\boldsymbol{U}$，逐行对矩阵进行消去，通过一次遍历，可以得到矩阵消减到不同维度时非零元的确切数目，从而大幅度的降低了运行时间。


x 10^4
2
1.8
1.6
1.4
1.2
1
0.8
0.6
非零元数量
计算时间最小
0
1000
2000
3000
4000
矩阵维度


x 10^5
5
4
3
2
1
0
非零元数量
计算时间最小
0
1000
2000
3000
4000
5000
矩阵维度

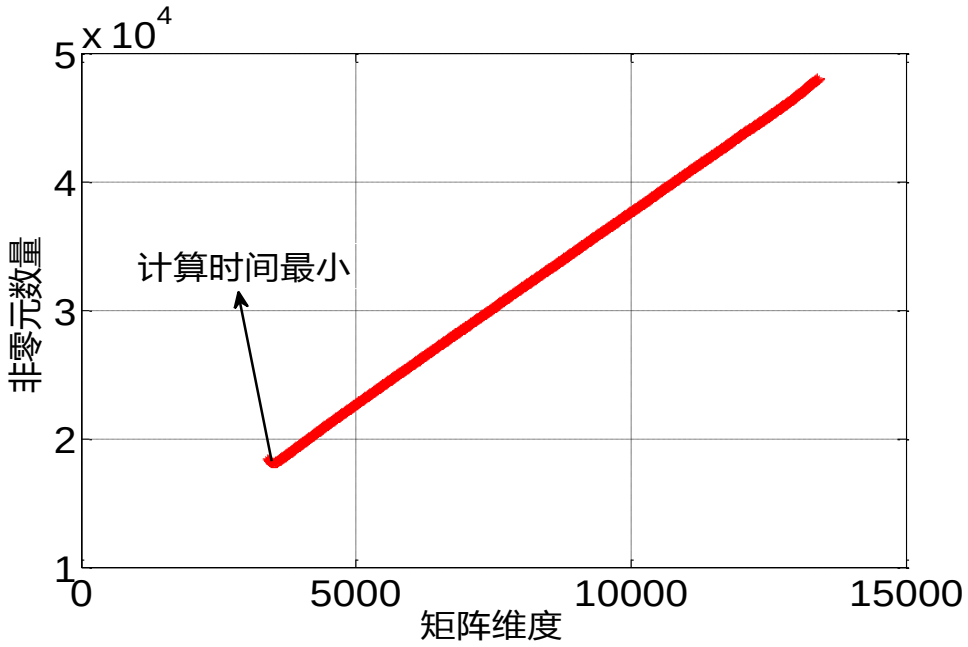


图 3.6 不同矩阵消去过程中非零元数量变化

3.4 根据约减结果预测矩阵处理时间

我们最终的目的是得到一个新的矩阵，使得处理这个矩阵的时间比原始矩阵时间要少，但是矩阵消减的过程中，非零元数目的变化没有规律，不同的矩阵消减过程中非零元数目变化差别很大（图 3.6），只有通过实际的消去才能得到矩阵非零元数量的变化情况。

图 3.6 展示了三个 RC 矩阵约减过程中非零元数量（y 轴）和维度（x 轴）的变化过程和最优消去的位置。可以看出最优消去的位置并不全是非零元最少或者维度最小的点，所以在得到了矩阵消去过程中维度与非零元的确切关系后也不能主观的得到最优消去点。假设一个矩阵在消减的过程中，维度降到初始的十分之一，但是非零元的数目增加到了初始的三倍，那么选择哪个做仿真速度更快一些是不确定的。

我们需要知道的是处理不同维度，不同非零元数目的矩阵所需要的时间，以及时间、维度、非零元三者之间的关系。对消去程序加以改进，将每消去一个节点后产生的新矩阵输出，用它进行相同的操作，比如 Cholesky 分解，统计出对应的运算时间。通过大规模的实验，用经验公式$f = \alpha + \beta * nz + \gamma * n$进行最小二乘拟合，解出$\alpha$、$\beta$、$\gamma$的值。通过这三个参数，在我们得到了矩阵维度和

非零元的关系后，就可以预测出矩阵在消减到不同程度时进行仿真所需要的运行时间，从而确定最优消去点，降低仿真用时。

下面给出本文提出的基于优化消去的模型降阶算法

算法 3.1. 基于优化消去的模型降阶算法

1. 将矩阵$\boldsymbol{A}$做 Camd 排序，把不可消去的节点置于矩阵对角线的右下角，数量为 s，取矩阵上三角部分为$\boldsymbol{U}1$；
2. 对更新后的矩阵$\boldsymbol{A}$做 Cholesky 分解，得到上三角阵$\boldsymbol{U}2$；
3. 将$\boldsymbol{U}1$、$\boldsymbol{U}2$中所有元素置为 1，$\boldsymbol{U} = \boldsymbol{U}1 + \boldsymbol{U}2$；
4. 将$\boldsymbol{U}$转化为 CSR 格式，生成向量$\boldsymbol{Ap}$、$\boldsymbol{Ai}$、$\boldsymbol{Ax}$；
5. For m = 1： size$(\boldsymbol{Ap}, 1) - 1 - $ s
6. $\boldsymbol{Ai}' = \boldsymbol{Ai}[\boldsymbol{Ap}(\mathrm{m}) + 1 : \boldsymbol{Ap}(\mathrm{m} + 1)]$;
7. 将$\boldsymbol{Ai}'$中的元素（即记录的该行非零元的列号）两两组合 记为(j, k)，提取其中第一个数字较小j＜k的组合方式；
8. 到$\boldsymbol{Ai}[\mathbf{Ap(j)} + \mathbf{1} : \mathbf{Ap(j + 1)}]$中去查找列号为k的位置l，即$\boldsymbol{Ai}(\mathrm{l}) = \mathrm{k}$；
9. 如果$\boldsymbol{Ax}$(l) = 1，则产生填入元；
10. 将$\boldsymbol{Ax}$中对应$\boldsymbol{Ai}$中遍历过元素的值置为 1；
11. End
12. 将更新完成后的向量$\boldsymbol{Ap}$、$\boldsymbol{Ai}$、$\boldsymbol{Ax}$转换为矩阵$\boldsymbol{U}3$；
13. $\overline{\boldsymbol{U}} = \boldsymbol{U}3 - \boldsymbol{U}1$，取$\overline{\boldsymbol{A}}$为$\overline{\boldsymbol{U}} + \overline{\boldsymbol{U}}^{\boldsymbol{T}}$的右下(n-m)×(n-m)部分，n 为$\overline{\boldsymbol{U}}$维度；
14. 将获得的维度与非零元用参数进行计算，确定运算时间最少的约减程度。

由于 RC 网络进行仿真时，需要考虑到电容矩阵$\boldsymbol{C}$和电阻矩阵$\boldsymbol{G}$，所以进行优化消去时也应当同时考虑到两者的影响，算法 1 中的矩阵$\boldsymbol{A} = \boldsymbol{G} + \boldsymbol{C}$，同时对$\boldsymbol{A}$进行了一定的处理，舍去了部分极小的数值。这样得到的经验公式同时反映了电阻和电容的影响。

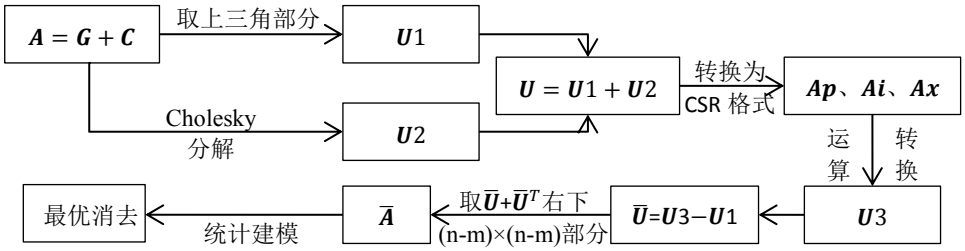


图 3. 7. 算法 3.1 流程

3.5 数值实验结果

所有实验均运行于 8 核 Intel(R) Xeon(R) CPU 的 Linux 工作站，CPU 的主频为 2.4GHz。程序由 Matlab 和 C 语言混合编程实现。

矩阵$\boldsymbol{T}1$是维度 93377、非零元数目 567297、不可消去的节点数为 8609 的 RC 矩阵，$\boldsymbol{T}2$ 是维度 446951，非零元数目 2456719，不可消去的节点数为 63764 的 RC 矩阵。按照不同的拆分方式将$\boldsymbol{T}1$分解成 89、45、34 个子矩阵，称为$\boldsymbol{T}11$、$\boldsymbol{T}12$、$\boldsymbol{T}13$。$\boldsymbol{T}2$拆分为 61 个子矩阵，称为$\boldsymbol{T}21$。

为了作为对比，我们在消去每个节点时先对矩阵进行一次 Camd 排序，并在排序的过程中利用约束将所有的端口节点排列到最后，然后再计算其一步的 Schur 补，此方法简称为 C&S 方法。以及本文的基于优化消去的模型降阶方法（Optimal Elimation based Reduction, 简称OER方法）。所用时间及加速比见表1。

表 1.不同 RC 矩阵用 C&S 及 OER 方法做预处理的时间对比

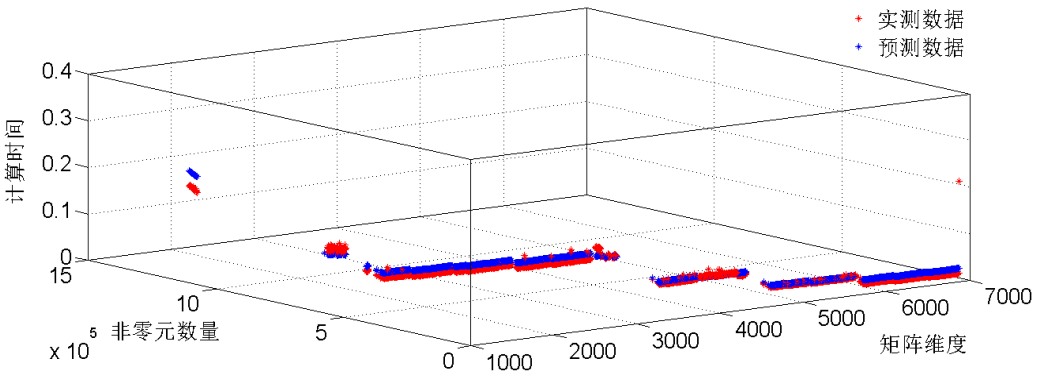
		C&S	OER	加速比
$\boldsymbol{T}1$	$\boldsymbol{T}11$	460s	3.9s	117X
	$\boldsymbol{T}12$	1536s	7.95s	193X
	$\boldsymbol{T}13$	3349s	13.29s	252X
$\boldsymbol{T}2$	$\boldsymbol{T}21$	Fail(out of memory)	288.3s	--

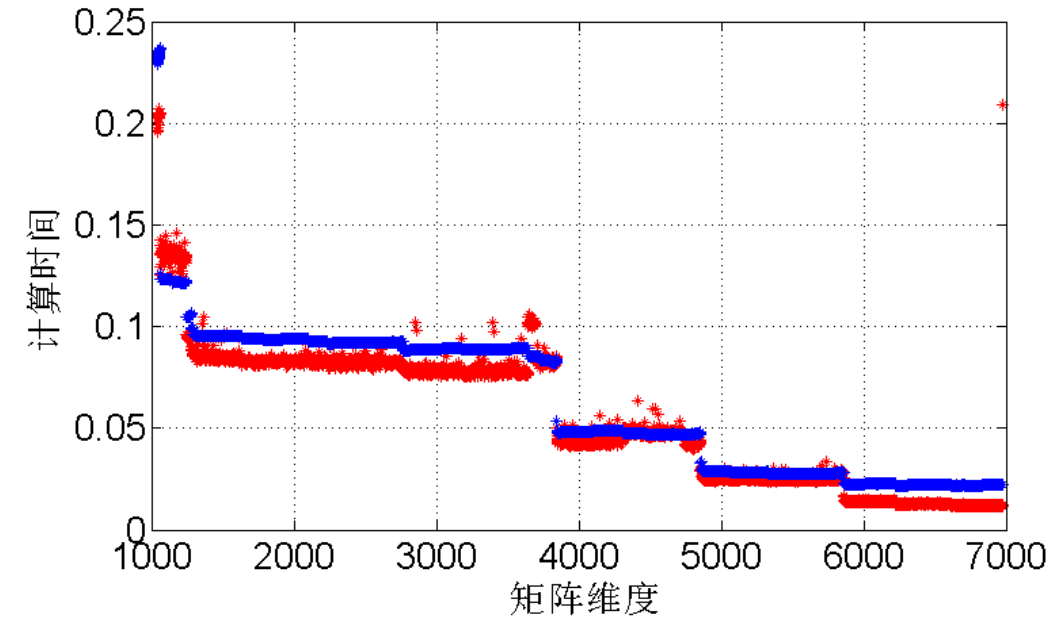
表 2 中的$\boldsymbol{T}3$是维度 22851，非零元 116629，不可消去的节点数 3404 的 RC 矩阵。依次选择不做处理（none），只保留节点维度（C&S），本文方法（OER）3 种方法处理后，进行 10000 次 Cholesky 分解所需要的时间。

表 2.不同预处理方法得到的矩阵做仿真的时间对比

		none	C&S	OER	加速比
$\boldsymbol{T}1$	$\boldsymbol{T}11$	2620s	1959s	1340s	1.96X
	$\boldsymbol{T}12$	3233s	4627s	1343s	2.41X
	$\boldsymbol{T}13$	3129s	7136s	1526s	2.05X
$\boldsymbol{T}2$	$\boldsymbol{T}21$	12869s	13976s	9339s	1.38X
$\boldsymbol{T}3$		2415s	29664s	445s	5.43X

根据拟合的结果，$\alpha = -5.5665 \times 10^{-4}$、$\beta = 2.0945 \times 10^{-7}$、$\gamma = 2.2567 \times 10^{-6}$，对预处理得到的矩阵维度和非零元计算可以得到仿真时间最少的矩阵消去程度。矩阵$\boldsymbol{T}2$ 分割出来的 61 个子矩阵的消去过程大部分和图 3.6 第 2 个矩阵相似，未处理已具有比较好的运算性能，所以和用矩阵直接运算对比，速度提升比较有限。$\boldsymbol{T}3$ 则不同，矩阵非零元数量经过消减后会减少，但是在最后会呈现极快的增加，如果采用 C&S 方法，得到的矩阵将是一个满阵，求解时间反而会大幅增加，本文所述方法对比 C&S 方法更好的确定了消减的程度，获得了 5 倍以上的加速比。





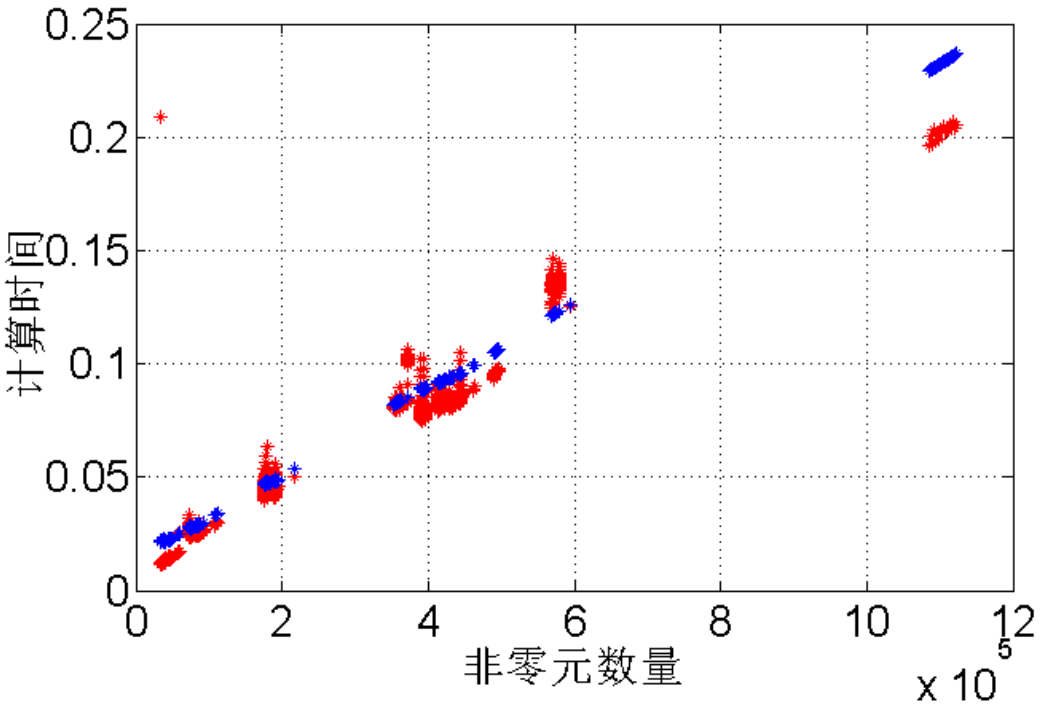


图 3.8 拟合结果比较

如果将矩阵预处理的时间与矩阵处理完毕后进行一次运算的时间相加，这个时间可能会比用原矩阵直接运算要长，但是在数模混合电路的后仿中，同样

的一个矩阵往往要求解很多次，甚至多达上万次，因此预处理所需的时间开销是值得的，它能使总的仿真时间显著缩短。

此方法的核心依然是节点消去法，没有方法误差，和其他的基于节点消去的方法得到的非零元数量及分布是相同的，降阶精度也是相同的，可参考文献[53]中的描述。使用通过统计建模得到的拟合参数对矩阵处理时间进行预测，平均误差大致在 10%左右，但是能准确的找到最佳的消去节点数，也就是说能较好的反映矩阵处理时间随着矩阵维度及非零元数量变化的趋势（见图 3.8）。

3.6 小结

本章针对大规模 RC 网络模型降阶，提出基于优化节点消去的模型降阶方法。该方法首先通过对于大规模的网络，快速得到剩余节点数与非零元的关系。其次采用统计建模的方法，得到网络求解时间与节点数和非零元的关系。通过这两个关系可以直接得到最优的节点消去策略。此方法可与文献中的 SIP 方法[53]结合得到复杂度最优的降阶网络。采用实际 IC 设置实验表明采用该策略可使得降阶网络的求解速度提高 2~5 倍。

第四章 总结与展望

4.1 总结

随着集成电路规模的不断扩大和特征尺寸的不断缩小，导致用于互连分析和电路仿真的 RC 网络规模越来越大，不仅直接处理非常困难，很多模型降阶技术也不再适用于规模如此大的问题。同时由于集成电路的结构也越来越复杂，同样对模型降阶技术提出了新的要求。本文对用于大规模稀疏 RC 网络的模型降阶问题进行了相关研究工作，主要研究内容及成果如下：

1.采用合适的稀疏矩阵排序技术，减少矩阵处理过程中的填入元。稀疏矩阵在处理的过程中，不可避免的会出现不同程度的 fill-in 现象，对于我们所处理的大规模稀疏 RC 网络，其稀疏性是我们希望能够保留的很好的性质，但是在处理过程中往往会破坏原始系统的稀疏性，产生非常稠密甚至是一个满阵，这样会大大的影响算法的效率，同时对存储空间的需求也很难满足。采用合适的稀疏矩阵排序技术，虽然无法避免填入现象的产生，但是能够能够在一定的程度上减少填入元，降低填入现象对算法的影响，从而改善问题的可计算性，同时提高算法的效率。经过实验，分别采用了 AMD、CAMD（强制最小度排序）等排序方法[54]，对比可知 CAMD 方法更适合用于电路后仿的大规模 RC 网络。

2.采用最优消去的策略，寻找降阶过程中效果最好的消去程度。采用节点消去法或者投影法对 RC 网络模型进行消去的过程中，RC 矩阵的非零元与维度之间是一个不规则变化的关系，根据处理对象的不同，差异可能会很大，有时因为填入现象的严重，使得维度降低后非零元数量上升很多。举例来说，对于一个维度降为原始系统十分之一,但是非零元数量却增长到原始系统 3 倍的系统，这样的降阶是否能够带来运算速度上的提升并不能确定。本文采取统计建模的方法，通过大规模的实验，拟合得到降阶系统的维度、非零元与处理时间的关系，利用这个关系，根据系统消去过程中维度与非零元的数量，得到降阶系统的最佳降阶程度。通过实验证明，采用此策略，能够使得到的降阶系统比用现有降阶方法得到的降阶系统，在处理时间上有 2 至 5 倍的加速。

4.2 展望

1. 本文中处理的矩阵，有的非零元随着维度的消减而减少，有的反之，还

有的变化不规则。我们希望的是非零元能够随着维度的减少有大致上减少的趋势，可以通过对原始系统进行预处理来改善这个问题。例如可以将原始系统进行分割，控制其特有的结构、形状等。但是这个问题在本文中并没有进行系统的研究，还停留在想法阶段，但是如果实现，那么对算法速度的提升将会有非常大的帮助。

2. 现有的矩阵排序技术基本上都是基于局部最优的技术；减少填入元的问题是一个 NP Hard 的问题，现在还没有理想的解决方法。但是如果能够更好地控制填入元的数量，对于大规模稀疏 RC 网络的模型降阶问题将是一个非常大的进步，无论从存储的角度还是从运算时间的角度都有非常大的帮助。

3. 由于本文采用的是统计建模的方法来进行优化消去，还需要增加更多、更大规模的例子来丰富和补充，这样才能更好、更准确、更全面的用于各个实际问题。

参考文献

[1] Semiconductor Industry Association, “International Technology Roadmap for Semiconducotrs,” [EB/OL]. http://public.itrs.net/, 2012

[2] M. Celik， L. Pileggi， and odabasioglu， IC Interconnect Analysis[M]， Norwell， MA: Kluwer， 2002.

[3] 杨华中，罗嵘，汪蕙. 面向微系统芯片的建模方法[M]. 北京: 清华大学出版社，2003.

[4] R. Achar and M. S. Nakhla， Simulation of high-speed interconnects， Proceedings of IEEE[J]， 2001， 89: 693-728.

[5] E. Ruehli， Equivalent Circuit Models for Three-Dimensional‘ Multiconductor Systems，IEEE Transactions on Microwave Theory and Techniques[J]， 1974， 22(3): 216-221.

[6] A. Devgan，H. Ji，and W. Dai，How to Efficiently Capture on-chip Inductance Effect: Introducing a New Circuit Element K[A]， in Proceedings of IEEE/ACM International Conference on Computer-Aided Design [C]， 2000.

[7] H. Ji， A. Devgan， and W. Dai， KSIM: A Stable and Efficient RKC simulator for Capturing on-chip Inductance Effect[A]， in Proceedings of IEEE/ ACM Asia and South Pacific Design Automation Conference[C]， 2001.

[8] Leon O. Chua and Pen-Min Lin， Computer-Aided Analysis of Electronic Circuits[M]，Prentice-Hall， 1975.

[9] D. Ling and A. Ruehli. Circuit Analysis， Simulation and Design-Advances in CAD for VLSI[M]， New York: Elsevier Science Publisher， 1987.

[10] C. W. Ho， A. E. Ruehli， and P. A. Brennan， The Modified Nodal Approach to Network Analysis， IEEE Transactions on Circuit and System[J]， 1975， 22: 504-509.

[11] B. D. Anderson and S. Vongpanitlerd， Network Analysis and synthesis[M]， Englewood Cliffs， NJ: Prentice-Hall， 1973.

[12] L. T. Pillage and R. A. Rohler， Asymptotic Waveform Evaluation for Timing Analysis，IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems[J]， 1990，9(4): 352-366.

[13] E. Chiprout and M. S. Nakhla， Analysis of interconnect networks using complex frequency hopping(CFH)， IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems[J]， 1995， 14(2): 186-200.

[14] P. Feldman and R. W. Freund， Efficient Linear Circuit Analysis by Pade Approximation via the Lanczos Process， IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems[J]， 1995， 14(5): 639-649.

[15] R. W. Freund and P. Feldman-n， Reduced-order modeling of large linear subcircuits via a bloke lanczos algorithm[A]， in Proceedings of IEEE/ACM Design Automation Conference[C]， 1995.

[16] E. J. Grimme， Kry1ov Projection methods for mode reduction[D]， Ph. D. dissertation Urbana-Champaign: Univ. Illinois， 1997.

[17] R. Freund and P. Feldmann， Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm[A]， in Proceedings of IEEE/ACM Interactional Conference on Computer-Aided Design[C]， 1996.

[18] R. W. Freund and P. Feldmann， The SyMPVL algorithm and its applications to interconnect simulation[A]， in Proceedings of IEEE/ACM International Conference on Computer- Aided Design[C]， 1997.

[19] E. J. Grimme， D. C. Sornsen， and P. M. Van Dooren， Mode reduction of State space system via an implicitly restarted Lanezos method， Numerical Algorithms[J]，1996， 12: 1-31.

[20] L. M. Silveira， M. Kamon，l. M. Elfadel， and J. White， Coordinate transformed arnoldi for generating guaranteed stable reduced- order models for RLC circuits[A]， In Proe. International Conference on Computer Aided Design[C]， 1996: 288-294.

[21] Zhaojun Bai， Peter Feldmann and Roland W. Freund， How to make theoretically Passive reduced-order models Passive in Practice[A]， IEEE Custom Integrated Circuits Conference[J]， 1998.

[22] Z. Bai，P. Feldmann，and R. Freund，Stable and Passive reduced-order models based on Partial Pade approximation via the Lanezos Process， ” Bell Laboratories， Lucent Technologies， Numerical Analysis Manuscript[J]， 1997: 3-10.

[23] R. W. Freund， Reduced-Order Modeling Techniques Based on Krylov Subspaces and Their Use in Cireuit Simulation. Numerical Analysis Manuscript[Jl， Bell Laboratories， 1998.

[24] Kevin J. Kerns and Andrew T. Yang， Preservation of Passivity during RLC network reduction via split congruence transformations[A]， IEEE/ACM Design Automation Conference[C]， 1997: 34-39.

[25] Kevin J. Kems and Andrew T. Yang， Stable and efficient reduction of large， multiport RC networks by Pole analysis via congruence transformations， IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems[J]， 1997， 16(7): 734-744，

[26] I. M. Elfadel and D. L. Ling， A bloke rational Arnoldi algorithm for multiport Passive model-order reduction of multiport RLC networks[A]， in Proceedings of International Conference on Computer-Aided Design[C]， 1997: 66-71·

[27] A. Odabasioglu， M. Celik， and L. Pileggi， Practical eonSiderati0nS for passive reduction of RLC circuits[A 〉， in Proceedings of International Conference on Computer-Aided Design[C]， 1999: 214-219.

[28] A. Odabasioglu， M. Celik， and L. Pileggi， PRIMA: passive Reduced-Order Interconnect Macromodeling Algorithm， IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems[J]， 1998， 17(8): 645-654.

[29] Qinjian yu， Janet Meiling L. Wang， Ernest S. Kuh， Passive multipoint moment matching model order reduction algorithm on multiport distributed interconnect networks， IEEE Transactions on Circuit and Systems-I[J]， 1999， 46(l): 140-160.

[30] R. W. Freund ， SPRIM: Structure-preserving Reduced-Order Interconnect Macromodeling[A] ， in Proceedings of IEEE/ACM International Conference on Computer-Aided Design[C]， 2004: 80-87.

[31] B. N. Sheehan， ENOR: Model order reduction of RLC circuits using nodal equations for efficient factorization[A]， in Proeeedings of IEEE/ACM Design Automation Conference[C]， 1999: 17- 21.

[32] H. Zheng and L. Pileggi， Robust and passive model order reduction for circuits containing susceptance elements[A]， in Proceedings of IEEE/ACM Interactional Conference on Computer- Aided Design[C]， 2002.

[33] Y. F. Su， J. Wang， X. Zeng， Z. Bai， C. Chiang， and D. Zhou， SAPOR: Second- Order Arnoldi Method for Passive Order Reduction of RCS Circuits[A]， in Proceedings of IEEE/ACM International Conference on Computer-Aided Design[C]， 2004: 74-79

[34] B. Liu， X. Zeng， and Y. F. Su， Block SAPOR: block second- order arnoldi method for Passive order reduction of multi-input multi-output RCS interconnect circuits[A]， in Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference[C]， 2005: 244-249.

[35] B. C. Moore，Principal component analysis in linear systems: Controllability，observability and model reduction， IEEE Transactions on Automatic Control[J]， 1981，35(1): 17-32.

[36] K. Glover， All optimal hankel-norm approximations of linear multivariable Systems and their I-error bounds， International Journal of Control[J]， 1982， 39(6): 1115-1193.

[37] R. H. Bartels and G. W. Stewart， Algorithm432: Solution of the matrix equation AX+XB=C，Commu. ACM[J]， 1972， 15: 820-826.

[38] K. Zhou， Frequency-weighted L_∞ norm and optimal Hankel norm model reduction，IEEE Transactions on Automatic Control [J]， 1995， 40(10): 1687-1699.

[39] P. Rabiei and M. Pedram， Model order reduction of large circuits using balanced truncation[A]， in Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference[C]， 1998.

[40] M. Kamon， F. Wang， and J. White， Generating nearly optimally compact m0dels from Krylov-subspace based reduced-order models， IEEE Transactions on Circuits and Systems11[J]， 2000， 47(4): 239-248.

[41] G. Wang， V. Sreeram， and W. Q. Liu， A new frequency-Weighted balanced truncation method and an error bound， IEEE Transactions on Automatic Control[J]， 1999， 44(9): 1734-1737.

[42] J-R. Li， F. Wang， and J. White， An efficient Lyapunov equation based approach for generating reduced-order models of interconnect[A]， in Proceedings of IEEE/ACM Design Automation Conference[C]， 1999

[43] P. Heydari and M. Pedram， Model reduction of variable-geometry interconnects using variational spectrally-weighted balanced truncation [A] ， in Proceedings of IEEE/ACM International Conference on Computer-Aided Design[C]， 2001.

[44] X. Zeng， D. Zhou， and W. Cai， An efficient DC-gain matched balanced truncation realization for VLSI interconnect circuit order reduction， Microelectronic Engineering[J]，2002， 60(1-2): 2- 15.

[45] J. R. Phillips， A statistical Perspective on nonlinear model reduction[A]， in Proceedings of Behavioral Modeling and Simulation Workshop[C]， 2003: 41-46.

[46] Joel R. Phillips and L. Miguel Silveira， Poor Man’s TBR: A Simple Model Reduction Scheme， IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems[J]， 2005， 24(l): 43-55.

[47] G. H.戈卢布，C. F. 范洛恩，矩阵计算[M]，北京:科学出版社，2002.

[48] 易大义，陈道琦. 数值分析引论[M].杭州:浙江大学出版社， 1998.

[49] P. Feldmann, “Model order reduction techniques for linear systems with large numbers of terminals,” Proceedings of DATE, pp. 944–947, 2004.

[50] P. Feldmann and F. Liu, “Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals,” Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on, pp. 88–92, Nov. 2004.

[51] P. Li and W. Shi, “Model order reduction of linear networks with massive ports via frequency-dependent port packing,” in DAC ’06: Proceedings of the 43rd annual conference on Design automation. New York, NY, USA: ACM, 2006, pp. 267–272.

[52] B. Sheehan, “TICER: Realizable reduction of extracted RC circuits,” Proceedings of IEEE/ACM International Conference on Computer Aided Design, 1999, pp. 200–203.

[53] Z. Ye, V. Dmitry, Z. Zhu, and J. R. Phillips, “Sparse implicit projection (sip) for reduction of general many-terminal networks,” Proceedings of IEEE/ACM International Conference on Computer Aided Design, Nov. 2008, pp. 736–743.

[54] Algorithm 837: AMD, An approximate minimum degree ordering algorithm, P. Amestoy, T. A. Davis, and I. S. Duff, ACM Transactions on Mathematical Software, vol 30, no. 3, Sept. 2004, pp. 381-388.

致　谢

首先衷心感谢我的导师喻文健副教授，本文的工作就是在他的悉心指导下完成的。在过去的三年里，喻老师在我的学习和生活上都给予了无微不至的鼓励和关怀，使我非常感动。同时他严谨踏实的治学作风、认真负责的工作态度也深深的感染了我，使我受益终身。

同时感谢微电子学研究所的叶佐昌老师在科研项目和论文写作方面给与的教导和帮助。蔡懿慈老师、周强老师也曾在课题和研究方面给予我指导与帮助，在此一并表示感谢。

感谢同课题组的张青青同学、胡超等同学，在与他们的讨论中我获益匪浅。还有实验室的孟畅、刘柳、李佐渭等同学，陪着我度过了精彩的三年时光。

最后，还要感谢我的家人与朋友，他们的关心与支持永远都是我积极进取的动力所在。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

程康，男，1983 年 2 月 13 日出生于河北省邢台市。

2001 年 9 月考入空军工程大学就读本科，2005 年 7 月本科毕业并获得工学学士学位。

2005 年 7 月就职于中国人民解放军空军第 93792 部队。

2009 年 8 月考入清华大学计算机科学与技术系，于软件研究所 EDA 实验室攻读工学硕士学位至今。

目前在审文章

[1] 程康，叶佐昌，喻文健，“基于优化消去的大规模 RC 网络模型降阶算法”，投稿至计算机辅助设计与图形学学报（已录用）