第3章

陈述式仿真模型的相容性分析

3.1 引言

基于方程的陈述式建模语言具有很多优点，但是也存在一个十分普遍的问题。那就是构建模型时，用户常常会不经意地遗漏方程或者多定义方程，即所谓的欠约束或过约束，致使仿真模型在结构上奇异，从而无法进行数值求解。解决欠约束、过约束问题的常用方法是补全遗漏的方程或者剔除冗余的方程。然而，复杂物理系统仿真模型包含的方程数目十分庞大，一个由几百个组件构成的模型通常可以产生上万个方程。当这样的复杂模型出现欠约束、过约束问题时，模型修正将是十分困难和费时的。但如果仿真建模平台的模型调试器能够采取适当策略自动地把与奇异有关的方程或变量限定到较小的范围内并提示给用户，那么模型的修正效率将显著提高。Peter Bunus 针对该问题提出了基于结构规则和语义规则的分析方法[113]。但该方法的调试能力有限，只适合分析简单模型，并且无法处理过约束与欠约束共存的问题。本章将在此基础上，利用模型的结构信息，采用基于组件的分析方式讨论模型的相容性问题。

3.2 方程系统表示图及相关概念

3.2.1 图论基本概念

定义 3.1 二部图

若无向图 $G=(V,E)$ 的顶点集合 V 能划分为两个子集 V_1 和 V_2，并满足：$V_1 \cap V_2=\varnothing$，$V_1 \cup V_2=V$。且对任意一条边 $e=(x,y)\in E$，均有 $x\in V_1$，$y\in V_2$，则称 G 为二部图。

定义 3.2 匹配、匹配基数、最大匹配

设 M 为图 $G=(V,E)$ 的边集 E 的任意子集，如果 M 中任意两条边均不相邻，则称 M 为图 G 的一个匹配。匹配 M 中边的数目称为匹配 M 的基数，记为 $|M|$。设 M 是 G 的一个匹配，如果不存在另一个匹配 M'，使得 $|M'|>|M|$，则称 M 为 G

的最大基数匹配，简称为最大匹配。

定义 3.3　自由顶点

设 v 是图 G 的一个顶点，M 为 G 的一个匹配，如果 M 中存在边与 v 关联，则称 v 是被 M 覆盖的。如果 v 没有被 M 覆盖，则称 v 为自由顶点。

定义 3.4　完美匹配

设 M 为图 G 的一个匹配，如果 M 覆盖了 G 中的所有顶点，则称 M 为图 G 的一个完美匹配。

定义 3.5　交错路径、可行路径

设 P 为图 G 中的一条路径，M 为 G 的一个匹配，$E-M$ 表示 G 中不属于 M 的边的集合，如果 P 的边交错地在 $E-M$ 和 M 中出现，则称 P 为交错路径，并将从 u 到 v 的交错路径简单表示为 $u \overset{=}{\longrightarrow} v$。如果交错路径 P 的起点和终点均没有被不属于该路径的匹配边覆盖，则称 P 为可行路径。

定义 3.6　传播域、先决域

在有向无环图上，从顶点 v 出发的所有路径上的顶点集合称为 v 的传播域。可达顶点 v 的所有路径上的顶点集合称为 v 的先决域。

3.2.2　结构关联矩阵

方程系统的一个基本结构属性是方程与变量的约束依赖关系，这种关系通常采用结构关联矩阵 $\boldsymbol{S}$ 来表示[106]。矩阵 $\boldsymbol{S}$ 的行对应于方程，列对应于变量。如果变量 j 出现在方程 i 中，那么元素(i,j)为 1；相反，如果变量 j 没有出现在方程 i 中，那么元素(i,j)为 0。对于式(3.1)给出的代数方程系统，其结构关联矩阵如式(3.2)所示。

$$\left.\begin{aligned} &e_1: f(x_1, x_2, x_3) = 0 \\ &e_2: f(x_1, x_2, x_4) = 0 \\ &e_3: f(x_3) = 0 \\ &e_4: f(x_3, x_4, x_5) = 0 \\ &e_5: f(x_4, x_5) = 0 \end{aligned}\right\} \tag{3.1}$$

$$\boldsymbol{S} = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} \begin{matrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{matrix} \tag{3.2}$$

3.2.3 结构关联矩阵的二部图表示

由于大规模方程系统通常具有非常明显的稀疏特性，故在其结构关联矩阵中存在大量为 0 的元素，为 1 的元素非常稀少。稀疏的结构关联矩阵可以用二部图表示。因为二部图同样可以清晰直观地表达方程和变量的约束依赖关系，而且图的存储结构紧凑，算法相当成熟和高效。

给定一个结构关联矩阵 $\boldsymbol{S}$，可以为其构造一个无向二部图 $G=(V_1,V_2,E)$。其中，V_1 中的顶点对应于矩阵 $\boldsymbol{S}$ 的行，表示方程；V_2 中的顶点对应于矩阵 $\boldsymbol{S}$ 的列，表示变量。如果矩阵 $\boldsymbol{S}$ 的元素 (i,j) 为 1，那么在方程 $i\in V_1$ 与变量 $j\in V_2$ 之间就存在一条边，该边表示变量 j 出现在方程 i 中。二部图抽象地表达了方程与变量之间的约束依赖关系。方程含有几个变量，方程顶点就有几条边与相应的变量顶点关联。反过来，变量出现在几个方程中，变量顶点就存在几条边与相应的方程顶点关联。由方程系统表示的数学模型实质上是一个约束系统，本书将方程系统对应的二部图称作该方程系统的约束表示图。图 3-1 给出了式(3.1)所示方程系统的约束表示图。

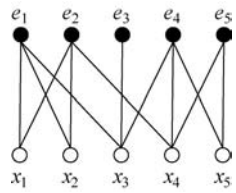

图 3-1 式(3.1)所示方程系统的约束表示图

3.3 仿真模型的相容性判定

定义 3.7 恰约束方程系统

设 E 为一个由 n 个方程组成的方程系统，当且仅当 E 满足以下条件时，称方程系统 E 是恰约束的。

(1) E 含有 n 个不同的变量；

(2) 在 E 中的每一个含有 $k(0<k\leqslant n)$ 个方程的子集中，至少存在 k 个不同的变量。

定义 3.8 结构相容、结构奇异

如果一个仿真模型 M 对应的方程系统 E 是恰约束的，则称仿真模型 M 是结构相容的。否则，称 M 是结构奇异的。

恰约束是方程系统可解的一个必要非充分条件，它只能从结构上而非数值上保证方程系统可解，也就是只能确保方程系统对应的仿真模型是结构相容的。本书只对结构相容问题进行探讨，不讨论数值相容问题。如无特别说明，书中提到的相容问题均指结构相容问题。非恰约束的方程系统可能是过约束的，也可能是欠约束的。在过约束系统中方程多于变量，在欠约束系统中，情况正好相反。一个方程系统也可能同时存在过约束和欠约束情形。

判断一个方程系统是否为恰约束系统的一个有效办法是对其中的方程和变量进行一对一配对，要求每个配对中的方程必须包含与其匹配的变量，并且一个方程只与唯一的一个变量形成配对，一个变量也只与唯一的一个方程形成配对。如果一个方程系统能够以上述方式将全部方程与变量形成配对，那么该方程系统是恰约束系统，否则不是。这种配对操作可以采用二部图上的最大匹配来实现[102]。基于方程系统的约束表示图，可根据如下规则判定一个方程系统是否为恰约束系统。

规则 3.1　设 M 为一个仿真模型，E 为 M 对应的方程系统，G 为 E 的约束表示图，如果 G 中存在一个完美匹配，那么方程系统 E 为恰约束系统，仿真模型 M 是结构相容的；否则，M 是结构奇异的。并且，若存在自由方程，则表明 E 是过约束的；若存在自由变量，则表明 E 是欠约束的；若二者均存在，则表明 E 中既存在过约束问题，也存在欠约束问题。

根据上述规则可以判定式(3.1)给出的方程系统是恰约束的，可以在其约束表示图中找到一个完美匹配，如图 3-2 所示，构成完美匹配的边由粗线表示。

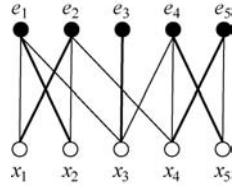

图 3-2　约束表示图 3-1 的一个完美匹配

判断一个约束表示图 G 是否存在完美匹配分两步完成。首先，找到图 G 的一个最大匹配 M；然后，判断 M 是否覆盖了图 G 的所有顶点。

定理 3.1[142]　若 P 为二部图 G 相对匹配 M 的一条增广路径，则由 M 和 P 可以获得一个新的匹配 $M\oplus P$，且 $|M\oplus P|=|M|+1$。

其中，符号⊕表示“异或”操作，即如果某条边 e 同时出现在匹配 M 和增广路径 P 中，则从 M 中去除 e；如果 e 只出现在匹配 M 或只出现在增广路径 P 中，则将 e 加入新的匹配中。

定理 3.2[142]　匹配 M 是图 G 的最大匹配的充要条件，是图 G 中不存在相对于匹配 M 的增广路径。

根据以上定理，通过在二部图中不断搜索增广路径，便可以获得二部图的一个最大匹配。在文献[142]中，Hopcroft 和 Karp 提出了一个最大匹配搜索算法。该算法的时间复杂度为 $O((m+n)\sqrt{m})$，其中 m 为图 G 的顶点数，n 为图 G 的边数。从时间复杂度上来说，该算法是迄今为止效果最好的最大匹配搜索算法。

由于相容性分析只注重方程系统的结构信息，即方程含有哪些变量，变量出现在哪些方程中，而不需要考虑方程和变量的具体表述形式，因此，在进行相容性判定时，我们将 DAE 问题当作代数问题看待，即将变量的导数当作变量本身。例如，把一阶 DAE 问题 $F(\dot{x},x,t)=0$ 当作代数问题 $F(x,x,t)=0$。对于式(3.3)给出的 DAE 系统，其约束表示图如图 3-3 所示。

$$\left.\begin{aligned} e_1&: v_x = \dot{x} \\ e_2&: m\dot{v}_x = -\frac{x}{L}F \\ e_3&: v_y = \dot{y} \\ e_4&: m\dot{v}_y = -\frac{y}{L}F - mg \\ e_5&: x^2 + y^2 = L^2 \end{aligned}\right\} \tag{3.3}$$

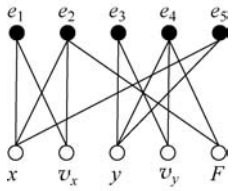

图 3-3　式(3.3)所示 DAE 系统的约束表示图

利用下面的定理可以将非恰约束方程系统的过约束或欠约束部分分离出来。

定理 3.3[142]　设 $V=V_1 \cup V_2$ 为二部图 $G=\{V_1,V_2,E\}$ 的顶点集合，集合 V 可划分为 3 个子集：D、A 和 C。其中，D 中的顶点没有被 G 的至少一个最大匹配覆盖；A 是集合 $V-D$ 的一个子集，A 中的顶点均至少与 D 中的一个顶点邻接；而 $C=V-A-D$。据此 3 个顶点集合，可将图 G 分解为 3 个子图：G_1、G_2 和 G_3。其中，$G_1=(C_1,C_2,E_1)$，$C_1=C\cap V_1$，$C_2=C\cap V_2$，$E_1=\{(u,v)\mid u\in C_1,v\in C_2\}$；$G_2=\{D_1,A_2,E_2\}$，$D_1=D\cap V_1$，$A_2=A\cap V_2$，$E_2=\{(u,v)\mid u\in D_1,v\in A_2\}$；$G_3=\{A_1,D_2,E_3\}$，$A_1=A\cap V_1$，$D_2=D\cap V_2$，$E_3=\{(u,v)\mid u\in A_1,v\in D_2\}$。

以上分解方法是由 Dulmage 和 Mendelsohn 在文献[170]中提出的，通常称作 DM 分解[170]。DM 分解的结果是唯一的，与二部图的最大匹配无关。其分解所得子图 G_1、G_2 和 G_3 中的 1 个或 2 个可能为空。DM 分解算法描述如下。

算法 3.1　DM 分解算法

输入：二部图 $G=(V_1,V_2,E)$，G 中的一个最大匹配 M。

输出：子图 G_1、G_2 和 G_3。

步骤 1：将图 G 中属于 M 的边替换为双向边，将不属于 M 的边 $\{(u,v)\mid u\in V_1,v\in V_2\}$ 定向为从 u 指向 v，生成有向图 $\bar{G}=(V_1,V_2,\bar{E})$。

步骤 2：找出顶点集 V_1 中所有没有被 M 覆盖的顶点，计算它们的传播域，并将传播域中的顶点从有向图中分离出来，得到子图 G_1。

步骤 3：找出顶点集 V_2 中所有没有被 M 覆盖的顶点，计算它们的先决域，并将先决域中的顶点从有向图中分离出来，得到子图 G_2。

步骤 4：在有向图 $\bar{G}$ 中除去子图 G_1 和 G_2，得到子图 G_3。

算法的时间复杂度为 $O(m+n)$，其中 m 为图 G 的顶点数，n 为图 G 的边数。

对某个方程系统的约束表示图执行 DM 分解，获得的子图 G_1、G_2 和 G_3 分别对应于该方程系统的过约束部分、欠约束部分和恰约束部分。如果分解结果中只有子图 G_3 不为空，那么表明该方程系统是恰约束的。蒋鲲基于 DM 分解提出了一个判断参数化模型的欠约束、过约束和完整约束性的图论算法[171]。图 3-4 给出了一个 DM 分解的例子，图中的方程系统同时存在过约束和欠约束问题。

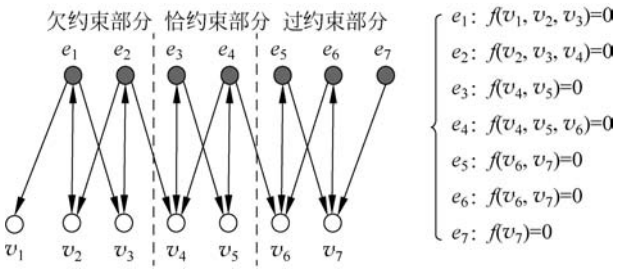

图 3-4　方程系统及其约束表示图的 DM 分解

3.4　奇异组件的识别

采用 Modelica 语言描述的物理系统的仿真模型是一种层次化的树状结构模型，它通过自底向上逐层聚合组件组合而成。对于一个复合模型来说，如果构成它的某个组件模型是奇异的，那么该模型必定也是奇异的。这意味着系统模型中任意层次上的一个组件奇异均会导致整个系统模型奇异。为此，可以根据模型的构成关系逐级地分析模型组件的相容性，来限定过约束问题或欠约束问题发生的大致范围。

然而，在 Modelica 模型中组件之间的连接通常是非因果的，即组件之间数据流的流向在定义连接时是不确定的，直到模型求解时才能确定。如果脱离组件的使用环境来单独分析一个组件的相容性，我们无法确定关联在该组件上的连接所产生的方程是否应该包含到该组件对应的方程系统中。

在此给出交流电动机模型的 Modelica 代码。

```
model Motor
  SineVoltage Vs(V = 220,freqHz = 50);
  Resistor Ra(R = 0.5);
  Inductor La(L = 0.1);
  EMF Emf;
  Inertia Jm(J = 0.001);
  Ground G1;
equation
  connect(Vs.p, Ra.p);
  connect(Ra.n, La.p);
  connect(La.n, Emf.p);
```

```
    connect(Emf.flange_b, Jm.flange_a);
    connect(Emf.n, G1.p);
    connect(Vs.n, G1.p);
end Motor;
```

在上述电动机模型中，组件电阻 Ra 和电感 La 之间存在连接 connect(Ra.n, La.p)，该连接会产生两个方程：$Ra.n.v=La.p.v$ 和 $Ra.n.i+La.p.i=0$。对于方程 $Ra.n.v=La.p.v$，如果它的求解方向最终被确定为基于 $La.p.v$ 求解 $Ra.n.v$，那么该方程应该包含到电阻 Ra 对应的方程系统中，此时 $Ra.n.v$ 为电阻 Ra 的输入变量，$La.p.v$ 为电感 La 的输出变量，它对电阻 Ra 来说可当作已知量对待。如果方程 $Ra.n.v=La.p.v$ 的求解方向最终被确定为基于 $Ra.n.v$ 求解 $La.p.v$，那么该方程应该包含到电感 La 对应的方程系统中。方程 $Ra.n.i+La.p.i=0$ 也是如此，它归属于哪个组件对应的方程系统取决于它的求解方向，即数据流的方向。

基于上述原因，在单独地分析一个组件的相容性时，为了切断组件之间的耦合关系，我们去除所有以连接方式施加在该组件的外部约束，并人为地添加一些虚构的方程来补偿失去的约束。在 Modelica 模型中存在两种类型的连接：因果连接和非因果连接[172]，针对这两类连接我们提出如下处理策略。

（1）对于因果连接，如图 3-5 所示，连接中数据的流向是明确的，从输入端连接器 A 流向输出端连接器 B。此时，我们为输入端连接器中的每个变量添加一个虚构的赋值方程，即为变量指定一个输入值，从而断开组件之间的耦合关系。

（2）对于非因果连接，如图 3-6 所示，连接中数据的流向没有明确指定，它可能从连接器 A 流向连接器 B，也可能从连接器 B 流向连接器 A，还可能两种情况均存在。此时，我们为连接两端的连接器 A 与 B 中的每一个流变量添加一个虚构方程，并使该虚构方程包含流变量所在连接器定义的所有变量。但如果连接器没有被连接，那么将直接让其中的流变量等于 0。

图 3-5　因果连接示意图

图 3-6　非因果连接示意图

因果连接是一种基于信号流的连接，此类连接具有明确的输入输出，对它的处理很简单，不再详加解释。

非因果连接是一种基于能量流的连接[172]，因为此类连接表达的是连接交汇点上的功率平衡、动量平衡、能量平衡或者质量平衡，即进出一个连接交汇点的功率、动量、能量或者质量是等值的。这正是 Modelica 语义规定在连接交汇点流变量之和为零、势变量值相等的依据。我们提出的为流变量添加虚构方程的策略也由此而来。一个通用的非因果连接器类可描述如下：

```
connector generic_connector
  Real e;
  flow Real f;
end generic_connector;
```

其中，e 代表势变量，f 代表流变量。假设 p、q 为该连接器类的两个实例，那么根据 Modelica 语义，连接 connect(p, q)可转化为两个方程：$p.e=q.e$ 和 $p.f+q.f=0$。根据这两个方程，可得到一个新的方程 $p.e\times p.f+q.e\times q.f=0$。若假设 $p.e\times p.f=C$，其中 C 表示某个常量，那么可得 $q.e\times q.f=-C$。

由此可见，通过基于方程 $p.e=q.e$ 和 $p.f+q.f=0$ 获得方程 $p.e\times p.f=C$ 和 $q.e\times q.f=-C$，我们可以切断非因果连接两端组件之间的耦合关系。由于进行相容性分析只关注方程系统的结构信息，并不关注每个方程的具体表达形式，故可以将方程 $p.e\times p.f=C$ 表述为隐式形式 $f(p.e,p.f)=0$。

对于陈述式面向对象建模而言，非因果连接必须能够确保在连接交汇点处的功率、动量或质量是平衡的，也就是进入连接点的功率、动量或质量必须等于流出连接点的功率、动量或质量。这一平衡原理也正是 Modelica 语言规定非因果连接中的流变量之和为零、势变量值相等的依据。故在设计物理连接器时，其流变量与势变量的选取虽然没有严格的限定，但要求基于该连接器的非因果连接必须能够满足连接机制的内在机理，即能够表达组件之间的动量平衡、功率平衡及质量平衡。

根据功率、动量和质量平衡原理，Modelica 语言已经为常见的几个领域定义了相应的连接器。在这些连接器中，势变量与流变量的乘积均表示动量、功率、质量或者能量。例如，在机械系统移动连接器中，流变量 f 表示力，势变量 s 表示位移，它们的乘积 $f\times s$ 为能量 E；在机械系统转动连接器中，流变量 τ 表示力矩，势变量 φ 表示转角，它们的乘积 $\tau\times\varphi$ 也为能量 E。在电路系统连接器中，势变量 v 表示电压，流变量 i 表示电流，它们的乘积 $v\times i$ 为功率 P。

综上所述，我们可以将为流变量添加的虚构方程 $f(p.e,p.f)=0$ 理解为一个关于功率、动量和质量的约束方程，即用该方程表示通过连接器的功率、动量和质量。

由于一个物理连接器中可能存在多对匹配的流变量与势变量。例如，机械多体连接器中就有 3 对匹配的流变量与势变量，分别表示 3 个不同坐标轴方向上的力与位移。故我们可以不失一般地将虚构方程表示为隐式形式 $f(e_1,e_2,\cdots,f_1,f_2,\cdots)=0$，其中 e_i 表示势变量，f_i 表示流变量。

此外，在物理连接器中，可能出现势变量个数多于流变量个数的情况。在多体系统建模中可能使用这样的连接器。若遇到这种情形，则需要为多余的势变量虚构一些额外的约束方程，并使虚构的每个方程均包含该组件的所有连接器中的所有势变量。设某个组件 C 有 r 个连接器，m_i 为第 i 个连接器的势变量数目，n_i 为第 i 个连接器的流变量数目，p 为组件 C 的变量数，q 为组件 C 的方程数，k 表示为多余势变量添加的虚构方程数。k 的具体值可根据以下 3 种情形确定。

(1) 如果 $p-\left(q+\sum_{i=1}^{r} n_i\right)<0$，令 $k=0$。

(2) 如果 $0 \leqslant p-\left(q+\sum_{i=1}^{r} n_i\right) \leqslant \sum_{i=1}^{r}(m_i-n_i)$，令 $k=p-\left(q+\sum_{i=1}^{r} n_i\right)$。

(3) 如果 $p-\left(q+\sum_{i=1}^{r} n_i\right) \geqslant \sum_{i=1}^{r}(m_i-n_i) \geqslant 0$，令 $k=\sum_{i=1}^{r}(m_i-n_i)$。

基于上述处理策略，我们可以将模型的任意一个组件从模型环境中脱离出来，通过添加虚构方程补全失去的外部约束，从而准确地判定该组件的相容性。假设现在要判定前面给出的电动机模型中的电阻组件 Ra 的相容性。Ra 是模型 Resistor 的实例，它有两个连接器 $Ra.p$ 和 $Ra.n$。我们为 $Ra.p$ 中的流变量 $Ra.p.i$ 添加虚构方程：$f(Ra.p.v, Ra.p.i)=0$，为 $Ra.n$ 中的流变量 $Ra.n.i$ 添加虚构方程：$f(Ra.n.v, Ra.n.i)=0$。这两个虚构方程与模型 Resistor 定义的 4 个方程一起构成 Ra 对应的方程系统，如表 3-1 所示。

表 3-1　电阻组件 *Ra* 对应的方程系统

方　　程	变　　量
$e_1: Ra.R * Ra.i = Ra.v$	$v_1: Ra.v$
$e_2: Ra.i = Ra.p.i$	$v_2: Ra.i$
$e_3: Ra.v = Ra.p.v - Ra.n.v$	$v_3: Ra.p.v$
$e_4: 0 = Ra.p.i + Ra.n.i$	$v_4: Ra.p.i$
$e_5: f(Ra.p.v, Ra.p.i)=0$	$v_5: Ra.n.v$
$e_6: f(Ra.n.v, Ra.n.i)=0$	$v_6: Ra.n.i$

在以上方程系统中，后两个方程是为补全缺失的外部约束而添加的虚构方程。该方程系统的约束表示图如图 3-7 所示，其中，由粗线边构成该图的一个最大匹配。由于最大匹配覆盖了图中的所有方程和变量，故它是一个完美匹配。根据规则 3.1 可知组件 Ra 是结构相容的。

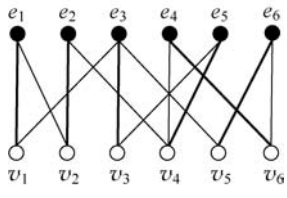

图 3-7　表 3-1 所示方程系统的约束表示图

下面对组件奇异的情况进行讨论。

现在修改组件 Ra 的模型 Resistor 的定义代码，在其中的方程定义部分添加一个多余的方程 $v=6$。此时，Resistor 已经变成了过约束模型，其实例 Ra 应该随之成为过约束组件。采用上述判定方法显然可以得出这一结论，因为 Ra 对应的方程系统将包含 7 个方程，而只有 6 个变量。

由此可见，采用上述处理策略可以准确地判定模型组件的相容性，不管其是相容还是奇异。然而，如果一个奇异组件是复合组件，即该组件也是由其他子组件通过连接方式聚合而成的，那么奇异组件识别过程还可以继续往下执行。为此，我们

给出如下定义。

定义 3.9　最小奇异组件

设 C 为某个模型的一个奇异组件，如果 C 满足以下两条件中的一个，则称 C 为最小结构奇异组件，简称最小奇异组件。

(1) C 为不可再分的简单组件；

(2) C 为复合组件，且 C 的所有子组件均不是奇异组件。

为了获取一个奇异模型的所有最小奇异组件，我们需要不断地分解识别出来的每个可以再分的奇异组件，直至遇到最小奇异组件。这一分解和识别过程可以当作一个子问题来对待，该子问题的求解如算法 3.2 所述。

算法 3.2　获取一个复合组件的奇异子组件

输入：一个复合组件 C。

输出：C 的所有奇异组件。

步骤 1：令集合 $S=\varnothing$，分解组件 C，将其子组件添加到 S 中。

步骤 2：取 $C'\in S$，从 S 中删除 C'。

步骤 3：为 C' 添加虚构方程，获得 C' 对应的方程系统 E。

步骤 4：为 E 构造约束表示图 G，并获取 G 的一个最大匹配 W。

步骤 5：若 W 不为 G 的一个完美匹配，输出奇异子组件 C'。

步骤 6：若 S 为空，退出；否则，转步骤 2。

获取一个奇异模型的最小奇异组件需要反复执行上述算法，整个识别过程构成一棵子问题树。基于该子问题树，采用深度优先方式逐个地求解子问题，便可以识别出一个奇异模型的所有最小奇异组件。因此，基于算法 3.2，我们可以构造一个最小奇异组件识别算法，算法步骤描述如下。

算法 3.3　最小奇异组件识别算法

输入：奇异模型 M。

输出：M 的所有最小奇异组件。

步骤 1：设 L 为一个组件链表，初始化 L 为空。

步骤 2：若 M 为简单模型，输出 M，退出；否则，令 $L=P(M)$。

步骤 3：若 L 为空，退出；否则，取 L 的链尾元素 C，从 L 中删除 C。

步骤 4：若 C 为简单组件，输出 C；否则，令集合 $Q=P(C)$。

步骤 5：若 Q 为空，输出 C；否则，依次将 Q 中的元素添加到 L 的链尾，转步骤 3。

其中，$P(M)$表示给定输入值 M 执行算法 3.2。很显然，在最小奇异组件识别过程中，识别算法利用了从编译器获得模型的结构信息，即根据模型的构成关系从上往下逐步细化分析范围，最终将分析范围定位在单个的组件上。本章后面部分将在此基础上展开进一步分析，尽可能地将错误根源限定在极少数的方程或变量上，进而提出相应的纠正方案供用户参考。