\*\*\*\*\*

工程力学》

\*\*\*\*

∦

文章编号: 1000-8608(2010)01-0001-08

# 高聚物注塑成型填充过程有限元分析并行迭代算法

### 李 征, 王希诚\*

(大连理工大学工业装备结构分析国家重点实验室,辽宁大连 116024)

**摘要:**进行了高聚物注塑成型填充过程并行数值仿真分析.首先给出问题的控制方程,然后用 Galerkin 法将其离散为有限元系统方程.发展了一个并行子结构迭代并行算法,该算法在有限元区域分解的基础上,将有限元节点分为子区域内部点、二子区域边界点和多子区域边界点,在此基础上实现了有限元方程的组集和求解的并行化,并研制了相应的程序.讨论了该算法的并行执行.最后给出两个注塑填充过程压力场分析的实例,数值算例表明所提方法有较高的并行计算效率,可以适应高聚物成型填充过程仿真分析的需要.

关键词:高聚物注塑成型;并行计算;有限元分析;迭代法 中图分类号:O34 文献标志码:A

## 0 引 言

20世纪70年代以后,电子计算机体系结构 发生了革命性变化,大规模并行处理已经成为现 代电子计算机发展的主流方向,为并行计算提供 了充分的硬件条件,与之相应的也就需要对应的 软件做支撑.有限元方程的组集和求解在很多大 型计算软件中都是计算速度的瓶颈,因此关于有 限元并行算法的研究是广受关注的一个热点研究 方向.有限元并行计算主要有两类算法,即直接并 行算法和迭代并行算法.

直接算法的计算量容易确定,在多工况情况 下实际计算量通常少于迭代算法,对于处理器数 量较少的情况,适合于使用直接算法<sup>[1]</sup>.对于多 工况结构分析问题,直接法中系数矩阵的三角化 分解只需计算一次,各工况分别回代求解即可,这 是迭代算法还未解决好的问题.而直接法往往难 于做到使处理器间的负载平衡.迭代算法具有较 好的并行性,如 EBE 的 PCG 法,只需要计算单元 刚度阵,不需要组装,而且各单元的计算量相当, 各处理器的计算任务容易平衡,减少了等待的时 间,并且一般需要的内存量小.但迭代法一般并行 粒度较细,当通信速度低时,很难提高效率.近年 来关于大型工程计算的并行算法研究也有很多, Fischborn 等<sup>[2]</sup>提出一种方法来提高 LU 并行预 处理的效率.刘耀儒等<sup>[3]</sup>将并行 EBE 技术应用到 拱坝-地基系统的大规模有限元数值计算中.20 世纪 90 年代以后,模具注塑的并行研究越来越受 到人们关注,如 Lee 等<sup>[4]</sup>提出了一种快速的低松弛 迭代算法用来求解大规模的 Laplace 方程,并且应 用到模具注塑的分析中,Yang 等<sup>[5]</sup>改进了模具填 充过程的 3D 并行模拟技术,得到了很好的效果.

本文在有限元子区域划分好的基础上,按照 多子区域边界、二子区域边界、子区域内部点的顺 序进行迭代以实现方程的并行求解,而从串行的 角度上看,其计算过程仅仅是改变了有限元节点 的编号顺序.

## 1 成型填充过程压力场计算

控制方程的求解主要包括 3 个阶段:压力场、 温度场和流动前沿位置的自动更替.沿用 FEM/ FDM/CVM<sup>[6]</sup>基本思想,型腔面内的压力场采用 有限元法,通过对时间和厚度方向的差分求解温 度场,并根据节点控制体积的充填状况更新流动 前沿.当熔体区域在任意充填时刻给定时,可以利

收稿日期: 2007-11-20; 修回日期: 2009-11-09.

基金项目:国家自然科学基金资助项目(10590354).

作者简介: 李 征\*(1982-),男,博士生;王希诚\*(1946-),男,教授,博士生导师, guixum@dlut. edu. en.

用压力边界条件通过求解压力控制方程而得到压 力场的分布.压力场的求解将在流动中面内进行, 二维流动平面被离散成三角形单元,单元内的压 力 *p*<sup>(a)</sup> 可以采用线性插值表示,

$$p^{(e)}(x,y) = \sum_{i=1}^{3} N_i p_i$$
 (1)

其中 N<sub>i</sub> 为面积坐标表示的线性插值函数<sup>[7]</sup>, p<sub>i</sub> 为 单元的 3 个节点压力值, 利用 Galerkin 加权有限 元法, 由压力控制方程和边界条件有

$$\int_{D} w \nabla \cdot S(p) \nabla p d\Omega + \int_{\Gamma^{q_1}} \overline{w}(S(p) \nabla p \cdot n - q_{\text{int}}) d\Gamma_1 + \int_{\Gamma^{q_2}} \overline{w}(S(p) \nabla p \cdot n) d\Gamma_2 = 0$$
(2)

式中:w、w 为加权函数;S(p)为流通率;q<sub>int</sub>为人 口点流率;D为计算区域;Γ<sup>q1</sup>为入口边界(流率边 界条件);Γ<sup>q2</sup>为各种模具边界(无渗透流率条件). 利用分部积分和散度定理有

$$\int_{D} w \,\nabla \cdot S(p) \,\nabla p \,d\Omega = \int_{D} \nabla \cdot (wS(p) \,\nabla p) \,d\Omega - \int_{D} (\nabla w \cdot S(p) \,\nabla p) \,d\Omega = -\int_{D} (\nabla w \cdot S(p) \,\nabla p) \,d\Omega + \int_{\Gamma} wS(p) \,\nabla p \cdot n \,d\Gamma = -\int_{D} (\nabla w \cdot S(p) \,\nabla p) \,d\Omega + \int_{\Gamma} wS(p) \,\frac{\partial p}{\partial n} \,d\Gamma = -\int_{D} (\nabla w \cdot S(p) \,\nabla p) \,d\Omega + \int_{\Gamma} wS(p) \,\frac{\partial p}{\partial n} \,d\Gamma = -\int_{D} (\nabla w \cdot S(p) \,\nabla p) \,d\Omega + \int_{\Gamma^{q_{1}}} wS(p) \,\frac{\partial p}{\partial n} \,d\Gamma_{1} + \int_{\Gamma^{q_{2}}} wS(p) \,\frac{\partial p}{\partial n} \,d\Gamma_{2} + \int_{\Gamma^{p}} wS(p) \,\frac{\partial p}{\partial n} \,d\Gamma_{3} \qquad (3)$$

其中 $\Gamma^{\mathfrak{p}}$ 为熔体前沿压力边界,结合式(2)和(3), 不失一般性,选择 $w = -\overline{w}$ 可以得到

$$\int_{D} (\nabla w \cdot S(p) \nabla p) d\Omega - \int_{\Gamma^{p}} wS(p) \frac{\partial p}{\partial n} d\Gamma_{3} - \int_{\Gamma^{q_{1}}} wq_{int} d\Gamma_{1} = 0 \qquad (4)$$

在 Galerkin 加权有限元法中,无渗透边界条件和入口边界条件为自然边界条件,选择合理的形函数,使在流动前沿(w = 0)并满足强制压力 边界条件 p = 0,并且 p、w 足够光滑,则有余量

$$R(p,w) = \int_{D} (\nabla w \cdot S(p) \nabla p) d\Omega - \int_{\Gamma^{q_1}} wq_{int} d\Gamma_1$$
(5)

对于平面三角形单元,不失一般性,令 $w_i = N_i$ ,则

$$R(p,w) = \int_{\Omega'} (\nabla N_i \cdot S(p) \nabla \sum_{j=1}^{3} N_j p_j) d\Omega' - \int_{\Gamma^{q_1}} N_i q_{int} d\Gamma = \int_{\Omega'} (\nabla N_i \cdot S(p) \nabla \sum_{j=1}^{3} N_j p_j) d\Omega' - \int_{\Gamma^{q_1}} N_i q_{int} d\Gamma = \int_{\Omega'} \left( \frac{\partial N_i}{\partial x} S(p) \sum_{j=1}^{3} \frac{\partial N_j}{\partial x} p_j + \frac{\partial N_i}{\partial y} S(p) \sum_{j=1}^{3} \frac{\partial N_j}{\partial y} p_j \right) d\Omega' - \int_{\Gamma^{q_1}} N_i q_{int} d\Gamma$$

$$(6)$$

写成矩阵形式为

$$\begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{22} & k_{23} \\ & & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}$$
(7)

其中单元刚度矩阵系数表达式为

$$k_{ij} = \int_{\mathfrak{A}^{\ell}} \left( \frac{\partial N_i}{\partial x} S(p) \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} S(p) \frac{\partial N_j}{\partial y} \right) d\mathfrak{Q}^{\ell}$$
(8)

组集压力场有限元总刚得到

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ab} & \mathbf{K}_{bb} \end{pmatrix} \begin{pmatrix} \mathbf{p}_{a} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_{a} \\ \mathbf{Q}_{b} \end{pmatrix}$$
(9)

式中:K 为刚度矩阵;K<sub>aa</sub> 为内部点分块阵;K<sub>bb</sub> 为 前沿点分块阵:p 为节点压力矢量;Q 为节点流率 矢量.相应边界条件为流动前沿节点的压力为零, 入口点的流率已知,而已充满节点静流率为零.

在求解过程中,将刚度矩阵、压力矢量以及流 率矢量分解,对于入口点和已充满节点,压力是未 知的,静流率为入口流率或零.而对于前沿节点, 压力为零而静流率是未知的.因此首先求解与入 口点和已充满节点相关的方程:

$$\boldsymbol{K}_{aa}\,\boldsymbol{p}_{a}\,=\,\boldsymbol{Q}_{a}\tag{10}$$

可以得到入口点和已充满节点的压力,然后回代 得

$$\boldsymbol{Q}_{\boldsymbol{b}} = \boldsymbol{K}_{ab} \boldsymbol{p}_{a} \tag{11}$$

得到前沿节点的静流率,由前沿节点的静流率来

确定新的前沿点位置和计算区域.由于 S 与压力 相关,方程组(10)可利用超松弛(SOR)迭代法求 解.

## 2 并行算法组织

超松弛迭代法在求解有限元方程中应用比较 普遍,可以按公式

$$p_{i}^{k+1} = p_{i}^{k} + \frac{\omega}{K_{ii}} \Big( Q_{i} - \sum_{j=1}^{i-1} K_{ij} p_{j}^{k+1} - \sum_{j=i}^{n} K_{ij} p_{j}^{k} \Big)$$
(12)

逐点进行迭代,从而不断更新各个节点的压力值, 直到收敛为止,一般线性方程的并行求解通过对 矩阵做行划分,结合即时通讯,将更新的值发送到 其余处理器中来完成[8].该方法可以适应于各种 线性方程组的求解,但其缺点在于荷载分布不均 衡,而且通信频率高,不能很好地满足大型有限元 方程求解需求.从式(12)中可以看出,只有当K<sub>ii</sub> 不为0时 $p_i$ 才会对 $p_i$ 的迭代产生直接影响,即在 每次更新节点 i 的压强时,仅仅需要提供所有与 节点 i 相邻的节点的当前压力值. 基于该特点, Ito 等「9」通过对有限元网格做红黑划分和区域分解 成功完成了并行计算的数据分割,其缺点是对于 具有多种单元的复杂有限元模型,红黑划分本身 就具有一定困难.为了进行并行计算,本文首先对 有限元区域做区域分解,将整个有限元计算区域 依照单元划分为互不重叠的 n 个子区域. 为了便 于迭代法组织,对节点进行如下分类:如果某节点 N的所有相邻单元分别属于子区域 $\{S_{i_1}, S_{i_2}, \dots, \}$  $S_{i_1}$ ,称节点 N 为子区域 $\{S_{i_1}, S_{i_2}, \dots, S_{i_n}\}$ 的点, 即 $N \in \{S_{i_1}, S_{i_2}, \dots, S_{i_n}\}$ ;如果节点N仅仅属于 一个子区域,即 $N \in \{S_i\}$ ,称该节点为子区域*i*的 内部点:如果N属于3个或3个以上子区域,则称 N 为多子区域边界点:设节点 N 仅属于两个子区 域,即 $N \in \{S_{i_1}, S_{i_2}\}, M$ 为与节点N相邻的边界

点, 若 M 也仅属于两个子区域, 即  $M \in \{S_{j_1}, j_n\}$  $S_{j_2}$ , 且 $\{i_1, i_2\} \neq \{j_1, j_2\}$ , 则 N 和 M 也为多子区 域边界点,否则 N 为二子区域边界点.由上面的 定义可以看出,即使节点N仅属于两个子区域,N 也不一定就是二子区域边界点.由此得出的分类 点具有如下性质:如果节点 N 和节点 M 为不同子 区域的内部点,那么在计算过程中,N和M的压 力值之间没有直接的影响;如果节点 N 和 M 分别 属于不同的二子区域边界,那么在计算过程中,N 和M的压力值之间也不会直接影响. 根据这些性 质,可以分别由不同的处理器迭代更新各子区域 的内部点和各二子区域边界上边界点的压力值, 从而实现并行计算.事实上,多子区域边界点的迭 代更新也可以再做更细的划分,但在实际问题中, 特别是二维问题中,多子区域边界点往往非常少, 故本文不做更细的划分.图1例示一个由4节点 单元剖分的有限元区域,共包括196个单元,分成 4个子区域、4个二子区域边界和1个多子区域边 界.



通过以上对有限元节点的划分可以将有限元 方程写成分块阵形式:

$$\begin{pmatrix} K_{1}^{I} & 0 & 0 & K_{11}^{IB} & \cdots & K_{1M}^{IB} & K_{1}^{IC} \\ 0 & \ddots & 0 & \vdots & & \vdots & \vdots \\ 0 & 0 & K_{N}^{I} & K_{N1}^{IB} & \cdots & K_{NM}^{IB} & K_{N}^{IC} \\ K_{11}^{BI} & \cdots & K_{1N}^{BI} & K_{1}^{B} & 0 & 0 & K_{1}^{BC} \\ \vdots & & \vdots & 0 & \ddots & 0 & \vdots \\ K_{M1}^{BI} & \cdots & K_{MN}^{BI} & 0 & 0 & K_{M}^{B} & K_{M}^{BC} \\ K_{11}^{CI} & \cdots & K_{NN}^{CI} & K_{1}^{CB} & \cdots & K_{M}^{CB} & K_{M}^{C} \\ \end{pmatrix} \begin{pmatrix} p_{1}^{I} \\ \vdots \\ p_{N}^{I} \\ p_{1}^{B} \\ \vdots \\ p_{M}^{B} \\ p_{C}^{C} \end{pmatrix} = \begin{pmatrix} Q_{1}^{I} \\ \vdots \\ Q_{N}^{I} \\ Q_{N}^{I} \\ \vdots \\ Q_{M}^{B} \\ Q_{M}^{C} \end{pmatrix}$$
(13)

上角标 *I*、B 和 C 分别表示内部点、二子区域边界 点和多子区域边界点.

SOR 迭代法的计算过程如图 2 所示,每一个 处理器都负责一个子区域的计算.首先在给定初 始压力值的情况下,由一个处理器根据 SOR 法对 式(14)做一次迭代更新,得出多子区域边界点的 结果,再将该更新后的结果 *p<sup>c</sup>* 发送给各个处理 器,各个处理器将接收到的 *p<sup>c</sup>* 代入式(15)中对所 有二子区域边界点做一次 SOR 迭代得出新的 *p<sup>B</sup>*, 然后将得出的 *p<sup>B</sup>* 代入式(16)完成对所有内部



图 2 并行求解计算过程



点的 SOR 迭代,如果当前得出的结果满足方程求 解精度,则停止计算,否则重复上述步骤.值得注 意的是在子区域数目比较大,且子区域划分不是 很规则的情况下,很难将二子区域边界点的计算 平均分配到各个处理器中,本文采用了并行计算 中常用的重复计算技巧,对所有以该节点作为边 界点的两个处理器各进行了一次,按照相同的先 后顺序(根据节点编号的大小)计算二子区域的边 界点以确保所有处理器中的边界点结果相同.

$$\boldsymbol{K}^{C}\boldsymbol{p}^{C} = \boldsymbol{Q}^{C} - \sum_{i=1}^{N} \boldsymbol{K}_{i}^{CI}\boldsymbol{p}_{i}^{I} - \sum_{i=1}^{M} \boldsymbol{K}_{i}^{CB}\boldsymbol{p}_{i}^{B} \qquad (14)$$

$$\boldsymbol{K}_{j}^{B}\boldsymbol{p}_{j}^{B} = \boldsymbol{Q}_{j}^{B} - \boldsymbol{K}_{j}^{BC}\boldsymbol{p}^{C} - \sum_{i=1}^{N} \boldsymbol{K}_{ji}^{BI}\boldsymbol{p}_{i}^{I}$$
(15)

$$\boldsymbol{K}_{j}^{I}\boldsymbol{p}_{j}^{I} = \boldsymbol{Q}_{j}^{I} - \boldsymbol{K}_{j}^{IC}\boldsymbol{p}^{C} - \sum_{i=1}^{M} \boldsymbol{K}_{ji}^{IB}\boldsymbol{p}_{i}^{B} \qquad (16)$$

## 3 算例与分析

#### 3.1 算例

**算例1** 图 3 所示为算例1 的有限元模型以及在应用5 个处理器并行求解下,填充结束时的区域分解模型,该模型包含2 242 个单元和1 225 个节点,最大子区域与最小子区域的单元数相差不超过当前计算区域单元数的1%.

计算结束后,应用 zmold 程序计算得出的压 力场分布如图 4(a)所示,图 4(b)为在计算结束后 应用 4 个处理器并行计算得出的压力场结果,可 以看出得出的结果满足计算要求.表1给出分别 应用 1~8 个处理器进行有限元分析时各阶段计 算的时间消耗和加速比.



Fig. 3 The model of the air-condition plate



(a) zmold 计算得出的压力场

(b) 4 个处理器计算得出的压力场

图 4 空调板压力场结果

Fig. 4 The pressure field of the air-condition plate

表 1	算例	1 有限	そ元各B	<b>}</b> 段计	算的問	亅间	消	耗和	加速	比
-----	----	------	------	-------------	-----	----	---	----	----	---

Tab. 1	The computing	time and speed	l up of the	Example 1	on different	number of	sub-processes
--------	---------------	----------------	-------------	-----------	--------------	-----------	---------------

处理器数	组集总刚时间/s	多子区域边界点 计算时间/s	内部点和二子区域 边界点计算时间/s	时间总和/s	加速比
1	53.90	0	37.600	91.525	1.000
2	28.50	0	20.100	48.624	1.894
3	19.80	0.091	14.700	34.591	2.651
4	14.70	0.380	11.300	26.380	3.472
5	12.99	0.610	9.811	23.411	3.913
6	10.40	0.880	8.560	19.840	4.615
7	8.95	1.240	7.310	17.500	5.233
8	7.95	1.290	6.510	15.750	5.812

**算例2** 手机壳模型如图5所示,该模型包含3730个单元和2062个节点,最大子区域与最小子区域单元数相差不超过当前计算区域单元数的1%.图5(a)为有限元模型,图5(b)为在填充

结束时应用5个处理器的区域分解模型.

计算结束后,应用 zmold 程序计算得出的压 力场分布如图 6(a)所示,图 6(b)为计算结束后应 用 4 个处理器并行计算得出的压力场结果,可以 看出得出的结果非常接近,满足计算要求.表2给 出了应用1~9个处理器进行有限元分析时各阶 段计算的时间消耗和加速比.



图 5 手机壳模型

Fig. 5 The model of the cell phone shell



图 6 手机壳压力场结果

Fig. 6 The pressure field of the cell phone shell

表 2 算例 2 有限元各阶段计算的时间消耗和加速比

处理器数	组集总刚时间/s	多子区域边界点 计算时间/s	内部点和二子区域 边界点计算时间/s	时间总和/s	加速比
1	70.90	0	52.90	123.830	1.000
2	37.59	0	29.10	66.719	1.858
3	27.10	0.125	22.30	49.525	2.501
4	21.10	0.178	17.50	38.778	3.020
5	17.10	0.400	13.80	31.300	3.954
6	14.70	0.542	12.10	27.342	4.539
7	12.60	0.640	10.89	24.130	5.124
8	11.70	0.770	10.50	22.970	5.389
9	10.40	1.120	9.32	20.840	5.935

#### 3.2 效率分析

表1和2分别列举了有限元总刚的组集时间,多子区域边界点、二子区域边界点和内部点的 计算时间.有限元总刚的组集时间与有限元模型 子区域划分的荷载平衡有关.由于多子区域边界 的计算为串行,二子区域边界的计算也是在两个 子区域的内部同时进行,有限元子区域的边界长 度也直接影响着计算效率的高低.图7描述了算 例1和2在应用不同数目处理器下的加速比(r) 情况.由图可见,两个算例各个处理器的效率大体 趋势上随着处理器数目的增加而降低,这是由于 多子区域边界点和二子区域边界点在整个有限元 节点中的比例随着子处理器数目的增加而增加, 对于有限元的区域分解来说这是不可避免的.

由算例1和2可以看出流动过程的数值模拟 本身计算量并不大,但在处理流动分析问题中,流 动过程计算本身往往要重复很多次,且应用迭代 法分析问题时,很难在比流动过程计算更高层次 上做并行计算,因此对流动过程的数值模拟做并 行计算还是必要的.



图 7 算例 1 和 2 在不同处理器数下的加速比 Fig. 7 The speed up of the two examples on different number of processor

## 4 结 语

本文研究了高聚物成型填充过程中压力场的 并行计算,按子结构迭代法进行了有限元分析.采 用对节点的分类来划分计算任务,而对节点的分 类主要依赖于子结构的划分.通过将有限元节点 划分为内部点、二子区域边界点和多子区域边界 点,实现了高聚物成型填充过程中压力场的并行 求解.实际算例表明本文提出的算法有较高的加 速比和计算效率,适宜于高聚物成型填充过程的 有限元并行计算.本文重点讨论了二维问题的并 行求解过程,所提出的算法也容易推广到三维问 题.求解三维问题时,由于边界点所占的比例可能 大于二维问题,需要将边界点做更加细致的划分.

## 参考文献:

- [1] 王希诚. 结构优化设计的并行计算方法[M]. 长春: 吉林大学出版社, 2000:204-234
- [2] FISCHBORN M, KUO-PENG P, SADOSWKI N, et al. LU parallel preconditioning with block intersection applied to FEM on computer clusters [C] // 12th Biennial IEEE Conference on Electromagnetic Field Computation. Hong Kong: CEFC, 2006
- [3] 刘耀儒,周维垣,杨 强. 三维有限元并行 EBE 方法
   [J]. 工程力学,2006,23(3):27-31
- [4] LEE C Y, LEE S M, OH J S, et al. Parallelization of a relaxation method to run on the Intel Paragon

[C] // Proceedings of the Conference on High Performance Computing on the Information Superhighway, HPC Asia'97. Washington D C: IEEE Computer Society, 1997:584-589

- [5] YANG W H, PENG A, LIU L, et al. Parallel true 3D CAE with hybrid meshing flexibity for injection molding [C] // SPE Annual Technical Conference — ANTEC. Boston: Society of Plastics Engineers, 2005: 56-60
- [6] HIEBER C A, SHEN S F. A finite-element/ finite-difference simulation of injection-molding filling

process [J]. Journal of Non-Newtonian Fluid Mechanics, 1980, 7(1):1-32

- [7] ZIENKIEWICZ O C. The Finite Element Method[M]. 3th ed. New York: McGraw-Hill, 1977
- [8]陈国良,安虹,陈 峻,等.并行算法实践[M].北 京:高等教育出版社,2004:531-533
- [9] ITO F, AMEMIYA N. Application of parallelized SOR method to electromagnetic field analysis of superconductors [C]// IEEE Transactions on Applied Superconductivity, 2004, 14(2):1874-1877

## A parallel iterative algorithm of finite element analysis for filling process of polymer injection molding

LI Zheng, WANG Xi-cheng\*

(State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian 116024, China)

**Abstract**: Parallel simulation of the polymer injection molding filling process is studied. The governing equations of the problem are given firstly, and then they are discretized into system of the finite element equations by means of Galerkin procedure. A domain decomposition parallel iterative algorithm is proposed. By classifying the nodes into sub-domain internal nodes, 2 sub-domain boundary nodes and multi-sub-domain boundary nodes, the assembling and solving of the system equation are parallelized, and a parallel program is developed. An implementation of this parallel method is discussed. Two numerical examples for the pressure analysis of the injection molding filling process are given. Numerical results show that the method gives high efficiency, and it is suitable for numerically simulating the injection molding filling process.

Key words: polymer injection molding; parallel computing; finite element analysis; iterative method