报　告　题　目	有限元单元开发报告
学　生　姓　名	曹业鸿
学　生　学　号	S230200249
专　业　班　级	2 3 0 2 班
学　院　名　称	机械与运载工程学院
指　导　老　师	王琥
学　院　院　长	丁荣军

日期：2024 年 1 月 2 2 日

目 录

一、开发说明......1
1、工具说明......1
2、开发流程......1
3、文件操作说明......1
4、开发单元介绍......1
二、 单元展示......2
1、 基本原理......2
（1）显式方程推导......2
（2）基本流程......2
2、 计算流程......4
3、 结果验证......5
（1）x 方向位移 U......5
（2）y 方向位移 V......6
4、 课程总结......7

一、开发说明

1、工具说明

ABAQUS_6.14、MATLAB_2018a、TECPLOT_360_EX_2017_R3。

2、开发流程

首先用 matlab 前处理程序生成初始模型的节点坐标信息，然后通过 matlab 的主程序进行计算，最后通过 matlab 的后处理程序输出计算结果(.plt 格式文件)到 Tecplot 中进行计算结果的云图显示并与 abaqus 进行比对。

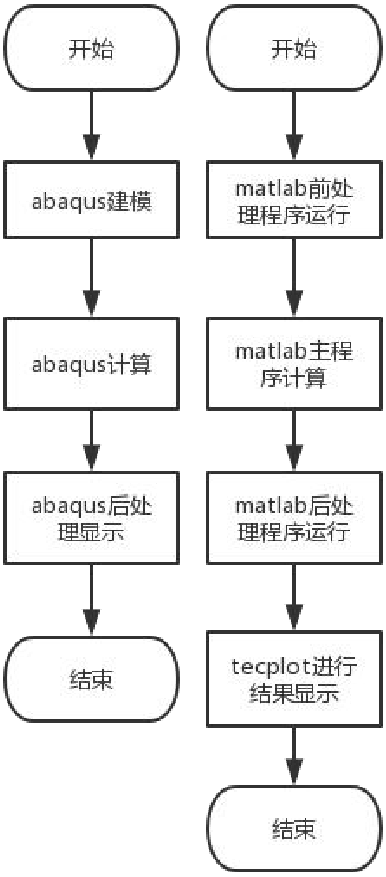


图 1、开发流程图

3、文件操作说明

explicit.m:主程序，包含前处理程序、计算程序和后处理程序。

angle.m、feglb.m、glb.m、glqd1.m、glqd2.m、glsb.m、jacob.m:自己编写的 matlab 子程序，里面都有功能注释。

result：输出结果文件夹，包含每一个输出步的.plt 文件。

4、开发单元介绍

CPS4：四节点四边形单元，计算方法采用等参单元全积分，计算格式采用显式格式。

二、单元展示

1、基本原理

平面四节点四边形单元的显式格式计算，采用时间迭代法，逐步求解。

隐式计算即构建单元的刚度矩阵，得到节点力与位移的关系，通过解线性方程组来解微分方程；而显式计算则是通过时间迭代来求解微分方程组，通过节点外力与内力之差，除以质量得到节点加速度，利用中心差分法根据加速度和时间步得到节点的速度和位移，然后根据新的节点位移更新节点内力，依次循环。

（1）显式方程推导

由平衡方程和边界条件得到变分形式如下：

$$\int_{\Omega}\delta \mathrm{u}(\frac{\partial\sigma_x}{\partial x}+\bar{b}_x-\rho\ddot{u})d\Omega-\int_{\Gamma q}\delta u(\sigma_x n_x-\bar{p}_x)d\Gamma=0$$

利用分部积分和散度定理且在位移边界上位移变分为零，则：

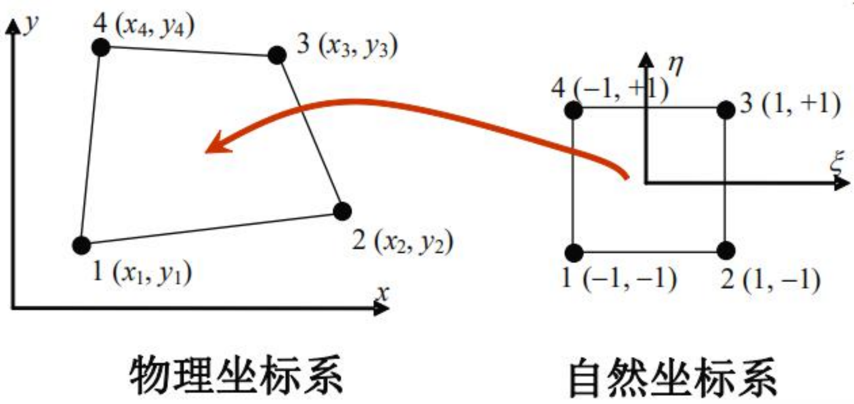
$$\int_{\Omega}\frac{\partial\delta u}{\partial x}\sigma_x d\Omega-\int_{\Omega}\delta u b_x d\Omega-\int_{\Gamma q}\delta u\bar{p}_x d\Gamma+\int_{\Omega}\delta u\rho\ddot{u}d\Omega=0$$

根据位移变分的任意性，上式可变形为：

$$\int_{\Omega}B^T\sigma_x d\Omega-\int_{\Omega}Nb_x d\Omega-\int_{\Gamma q}N\bar{p}_x d\Gamma+\int_{\Omega}N\rho\ddot{u}d\Omega=0$$

（2）基本流程

物理坐标下的任意四边形单元向自然坐标下的标准矩形单元转换，要满足两者用同样的节点和形函数进行插值。



自然坐标下节点的位移场满足：

$$u=\alpha_1+\alpha_2 x+\alpha_3 y+\alpha_4 xy$$

$$v=\beta_1+\beta_2 x+\beta_3 y+\beta_4 xy$$

代入（1,1）、（1，-1）、（-1,1）和（-1，-1）四个节点坐标，得到：

$$u=N_1u_1+N_2u_2+N_3u_3+N_4u_4$$

$$v=N_1v_1+N_2v_2+N_3v_3+N_4v_4$$

$$N_1=0.25(1-\xi)(1-\eta)$$

$$N_1=0.25(1+\xi)(1-\eta)$$

$$N_1 = 0.25(1+\xi)(1+\eta)$$

$$N_1 = 0.25(1-\xi)(1+\eta)$$

由形状函数矩阵得到应变矩阵 B，i=1、2、3、4 表示四边形单元的四个节点，J 表示雅各比矩阵：

$$\mathbf{B}(\xi,\eta)=\begin{bmatrix}\frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x}\end{bmatrix}\begin{bmatrix}N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0\\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4\end{bmatrix}$$

$$\begin{Bmatrix}\frac{\partial N_i}{\partial x}\\ \frac{\partial N_i}{\partial y}\end{Bmatrix}=\mathbf{J}^{-1}\begin{Bmatrix}\frac{\partial N_i}{\partial \xi}\\ \frac{\partial N_i}{\partial \eta}\end{Bmatrix}$$

$$\mathbf{J}=\begin{bmatrix}\frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi}\\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta}\end{bmatrix}\begin{bmatrix}x_1 & y_1\\ x_2 & y_2\\ x_3 & y_3\\ x_4 & y_4\end{bmatrix}$$

由应变矩阵和材料矩阵 D 得到单元的节点内力 f^{in}：

$$\mathrm{f}^{\mathrm{in}} = \int_\Omega \mathrm{B}^{\mathrm{T}} DBd\Omega$$

用高斯数值积分计算上式，得到：

$$\mathrm{f}^{\mathrm{in}} = \int_{-1}^{1}\int_{-1}^{1} B^T DBh\,|J|\,d\xi d\eta$$

单元的节点外力 f^{ex} 为：

$$f^{ex} = \int_\Omega N\rho b d\Omega + (N\bar{t})\big|_{\Gamma_t} = \text{体力+边界力}$$

N 为形状函数矩阵，$\bar{t}$ 为力边界单位法线上的应力，b 为单位质量力，即体力。

根据节点的质量 m 得到单元节点的加速度 a：

$$\mathrm{a} = (\mathrm{f}^{\mathrm{ex}} - \mathrm{f}^{\mathrm{in}})/\mathrm{m}$$

由中心差分法可得单元节点的速度和位移：

$$v^{n+1/2} = v^{n+1/2-\alpha} + \alpha\Delta t a^n$$
$$\alpha = 1/2, n=0; \alpha = 1, n>0$$

$$u^{n+1} = u^n + \Delta t v^{n+1/2}$$

2、计算流程

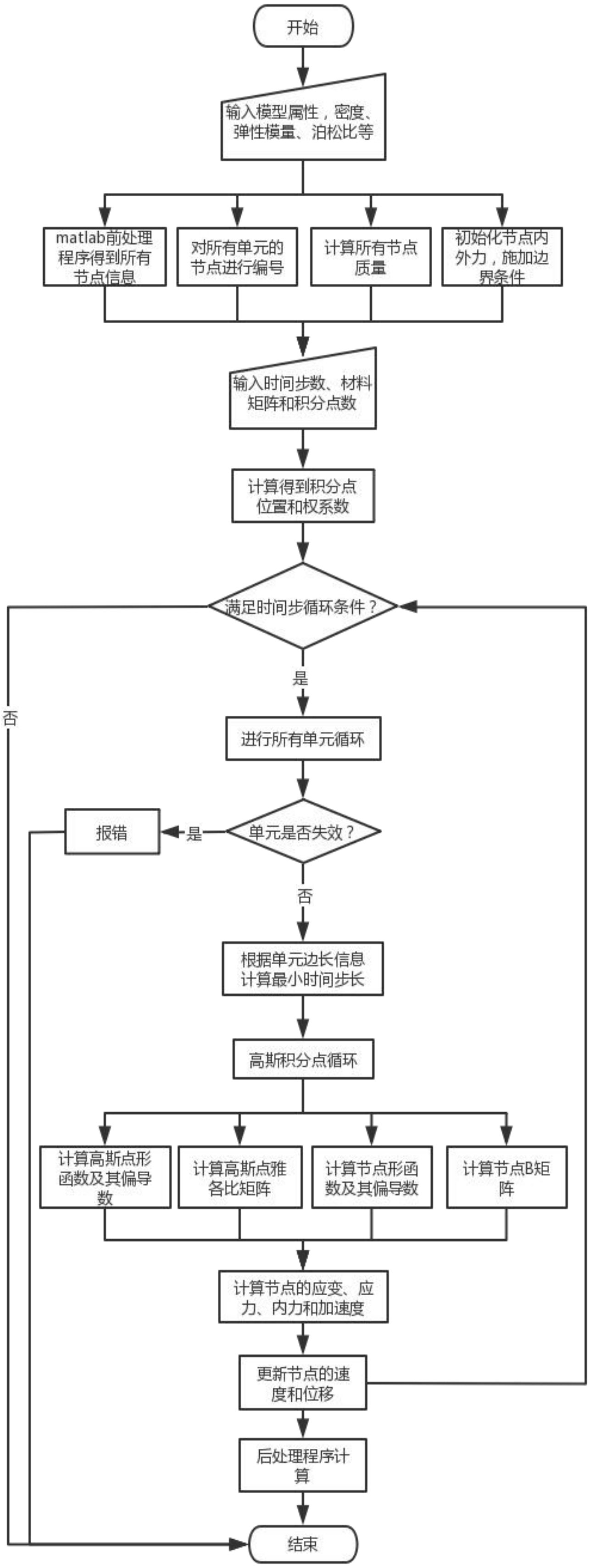


图 2、计算流程图

3、结果验证

（1）x 方向位移 U

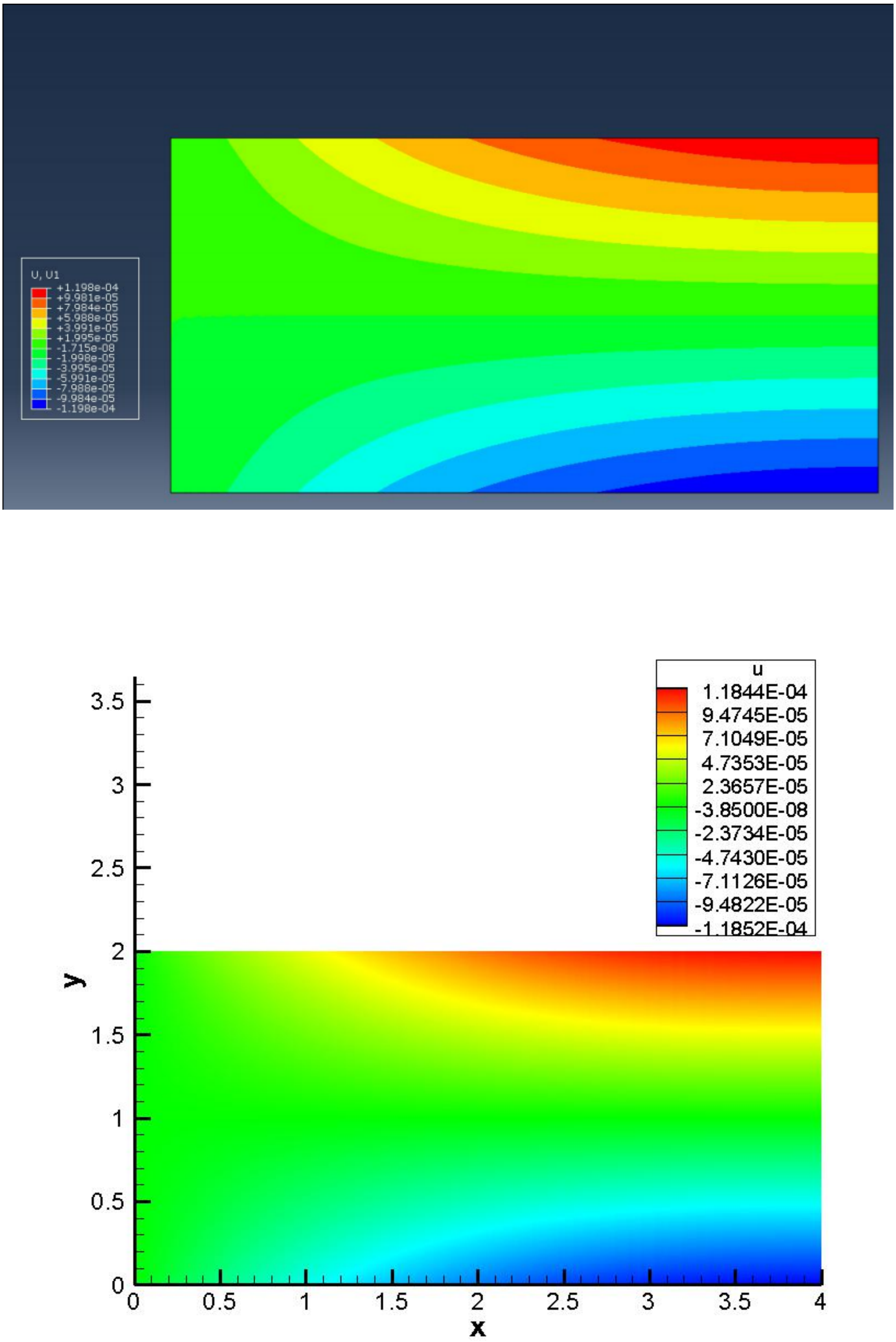


图 3、x 方向位移 abaqus 和 matlab 计算结果云图

表 1、x 方向位移误差表

	最大值	最小值
Abaqus 结果	1.198e-5	-1.198e-5
Matlab 程序结果	1.184e-5	-1.185e-5
误差	1.17%	1.09%

（2）y 方向位移 V

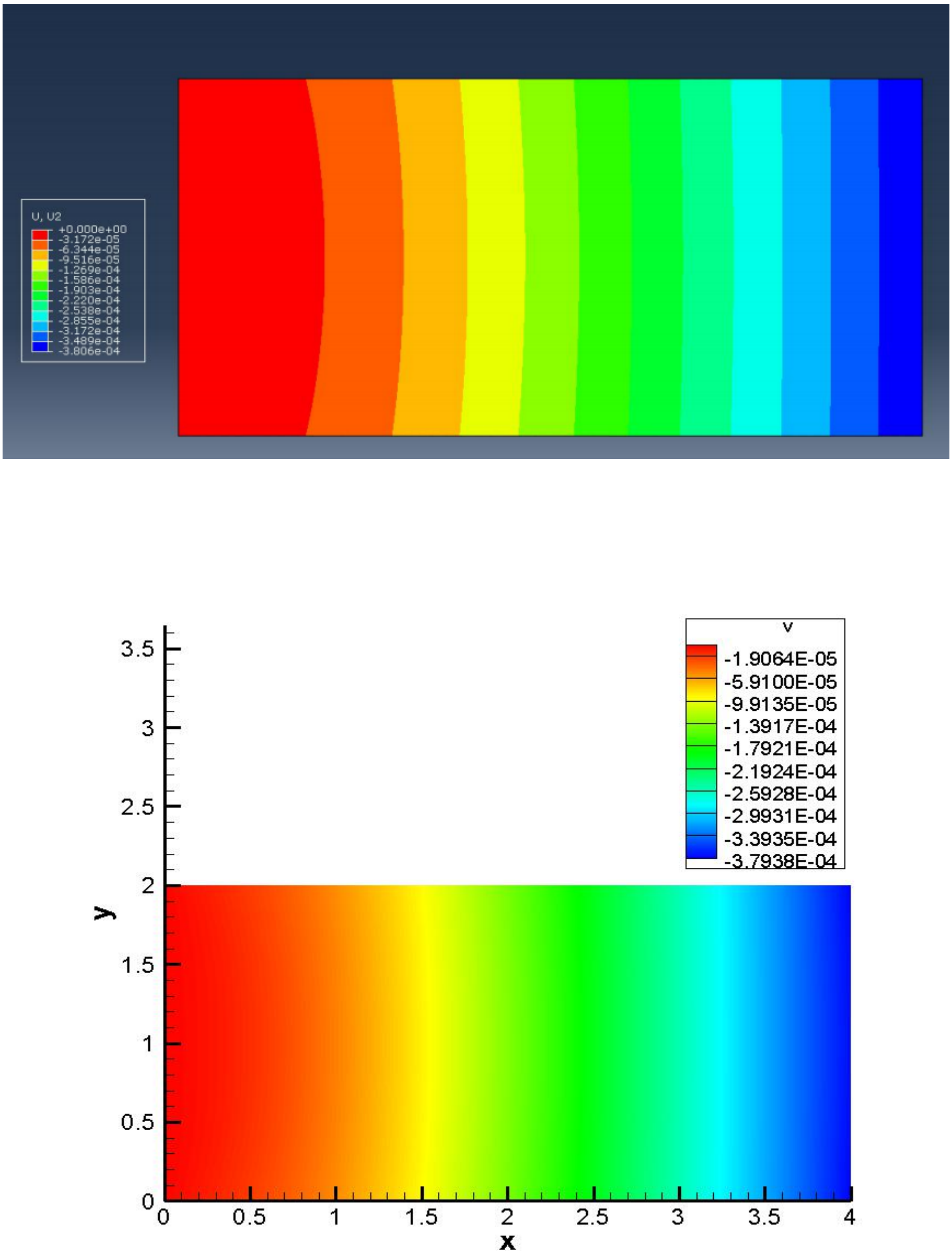


图 4、y 方向位移 abaqus 和 matlab 计算结果云图

表 2、y 方向位移误差表

	最大值	最小值
Abaqus 结果	0	-3.806e-4
Matlab 程序结果	0	-3.794e-4
误差	0	0.315%

4、课程总结

首先，我要感谢王琥老师这八周的的细心讲解和指导，让我对有限元的基本理论有了一定的了解和认识，也对此方向产生了些许兴趣，其次，我要感谢很多同学的帮助，让我能够顺利完成这次四节点四边形单元的程序设计。

通过这门课程，我了解到了有限元网格的发展之必要性及其实用性，它可以通过仿真解决很多工程上需要耗费大成本才能实现的实验项目，作用很大。有限元最近几十年发展迅速，技术也逐渐趋于成熟，目前主流的有限元商业软件如 Abaqus、Ansys、Ls-Dyna 等在各大领域都发挥重要作用。同时，有限元技术还有很多问题如复杂单元精度、网格剖分精度、几何清理、网格自适应和流固耦合等问题都是需要大家共同努力去解决的。最后，希望国人能够努力开发出属于我们自己的商业软件，不再受外国相关技术的制约。