

# FEPG.NET V6.0

## 白皮书

为您带来有限元计算的超强创新能力

# 摘要

本白皮书提供飞箭 FEPG.NET V6.0 的概述，并详细叙述了它的系统要求、技术和优点。在描述系统体系结构的同时，还介绍了 FEPG.NET V6.0 的关键组件和优点，包括多物理场仿真、管理和部署、OpenMP、MPI、编译器和解算器。

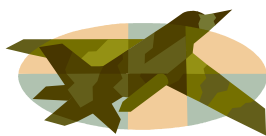
这是一份预备文档，在文中所述软件的最终商业版本发布之前，可能会有重大改动。

本文包含的信息仅代表飞箭公司在本文发布时对所讨论问题的观点。由于飞箭公司必须顺应不断变化的市场条件，这些信息不应视为飞箭公司方面的承诺，同时飞箭也不能保证本文发布之后其他任何信息的准确性。

本白皮书仅供参考。飞箭对本文档中的信息不提供任何形式的(包括明示或暗示的)保证。

用户有责任遵守所有适用的版权法。除版权法所赋予的权利外，未经飞箭明确的书面许可，任何人不得将本文复制、存储或引入检索系统，或是以任何形式或通过任何手段(电子、机械、影印、录制或其他方式)或为任何目的传播本文的任何部分。

所有其他商标均属其各自所有者的财产。



FEPG.NET V6.0 是面向未来的有限元平台，是工作组规模以上团队进行有限元教学、研究及工程设计最为理想的解决方案。用最先进的技术武装头脑，用最快捷的方法自主创新，用最低的成本来模拟世界，FEPG.NET 可以做到。

	各种 PC	各种并行机及各种网络互连方式	Linux/ Windows	32 位/ 64 位	局域网/ 万维网
FEPG.NET	✓	✓	✓	✓	✓

# 目录

1. 什么是 FEPG.NET V6.0?	4
1.1. 背景	4
1.2. FEPG.NET V6.0 体系组成	4
2. FEPG.NET V6.0 的新特性	6
2.1. 生成系统的新特性	6
2.2. 前后处理的新特性	6
2.3. 界面的新特性	6
2.4. 文档的新特性	7
3. 硬件要求	8
3.1. 服务端	8
3.2. 客户端	8
3.3. 计算端	8
4. 操作系统及支撑环境	9
4.1. 服务端	9
4.2. 客户端	9
4.3. 计算端	9
5. FEPG.NET V6.0 的绝对优势	11
5.1. 最本质的多物理场耦合	11
5.2. 最方便的部署与管理	11
5.3. 最创新的应用成果	12
5.4. 最廉价的版税	错误！未定义书签。
6. 结束语	12
7. 附录	12
7.1. OpenMP	12
7.2. MPI	13
7.3. 编译器和数学库	13



飞箭将一如既往地无偿为来自全球的非商用有限元研究和应用人员提供 FEPG.NET 的使用，我们也会非常乐意为大家提供技术支持并接受各种意见和建议。

详细请进 <http://www.fegensoft.com>

# 1. 什么是 FEPG.NET V6.0?

火箭自 2000 年起就开始支持互联网有限元计算。用户通过广域网或局域网随时随地都可以进行有限元程序或软件的开发。FEPG.NET V6.0 是 IFEPG V5.2 的升级版本，并随着其发布，火箭将高性能计算的超级计算能力引入了互联网领域。



---

并行计算是高性能计算的基础，也是高性能计算的核心。但并行软件的研制和开发是高性能计算的瓶颈问题。支持多内核 PC 与集群系统的 FEPG.NET V6.0 则可以从根本上解决 CAE 领域的并行软件开发的难题，成为自主创新的引擎。

---

## 1.1. 背景

有限元程序自动生成系统(Finite Element Program Generator, FEPG) 是工程和科学计算领域领先的有限元问题求解环境(Problem Solving Environment, PSE)。研发始于 1983 年，由中国科学院数学所梁国平研究员发明。系统采用组件化设计方法，应用自动生成技术和公式库技术，可以将用户应用有限元语言编写的微分方程表达式和算法文件编译成 Fortran 源代码，供有限元软件开发和科学与工程计算之用。



概括而言，问题求解环境的基本特征有：面向科学和工程计算中的一类问题；使用用户熟悉的方式描述问题；易用、结果可靠、高性能；组件化、模块化、罐装（用户无需知道细节）；算法参数化；领域知识管理；多层抽象。PSE 基本的设计原则可以概括为三点：人为中心，组件为基础，交互的方式。PSE 通常的基本结构包括三个层次：应用开发框架,软件基础设施，算法与系统基础。软件基础设施和算法与系统基础提供一整套翻译、编译、运行工具集和环境，为上层应用提供基础。

## 1.2. FEPG.NET V6.0 体系组成

FEPG.NET V6.0考虑到了各种用户的需求，用户可以根据自己的需要进行部署和应用。

## ◆ 基础设施

### ☞ SRL.NET

串行程序生成系统。本模块可将用户编写的有限元文件生成串行有限元计算程序，程序由五个计算组件组成，组件以文件名为接口。计算程序的数据交换通过磁盘文件进行，数据结构简单，不依赖专门数据库。程序经编译后可在任何一台PC或服务器上运行。

### ☞ OMP.NET

多线程并行程序生成系统。本模块可将用户编写的有限元文件生成基于OpenMP的并行有限元计算程序。程序由五个计算组件组成，组件以文件名为接口。计算程序的数据交换通过磁盘文件进行，数据结构简单，不依赖专门数据库。程序经编译后可在任何一台双核或多内核PC、或SMP并行机上运行。编译器目前支持Intel Fortran Compiler V8.1以上版本，程序运行过程中依赖 Intel MKL V9.0以上版本。

### ☞ MPI.NET

分布式并行程序生成系统。本模块可将用户编写的有限元文件生成分布式并行有限元计算程序。程序采用消息传递编程模型，计算由一个或多个进程构成，进程间通过调用库函数发送和接收消息来完成通信。程序具有高度可移植性，不需要特殊硬件来支持它的有效执行，能够运行于任何并行机。

**FEPG.NET 的 MPI.NET 模块对内存分配和通信次序等消息传递编程的所有难点都进行了自动处理，用户无需对此细节多加考虑。**

<b>FEPG.NET</b>  <b>V6.0</b>	<b>1</b> 组 FEPG 文件
	<b>3</b> 条命令
	<b>3</b> 种程序源代码
	<b>3</b> 分钟全部搞定



## 2. FEPG.NET V6.0 的新特性

### 2.1. 生成系统的新特性

- 增加了MFS多物理场耦合系统，使多场耦合计算更加直接、完善、灵活
- 编写有限元文件 MFI 文件实现多场、多种单元类型、多种材料属性。物理问题的描述更加简单、人性化
- 支持于单个物理场有多个微分方程表述
- 支持多线程并行，可以在双核或多内核 PC 上实现并行计算
- 全面支持 Intel Compiler 和 Intel MKL，程序运算速度得到成倍提升

### 2.2. 前后处理的新特性

- 支持 64 位计算和 UTF-8 编码
- 增加了多种网格剖分准则和算法
- 支持将合并在一起的体分开，支持多个面求交
- 更容易读取 VDA 和 IGES 文件（更快，且需要更少的内存）
- 支持参数化创建曲面
- 可以导入 Femap v1.0 的二进制结果文件
- 可以在两点之间创建剖视图

### 2.3. 界面的新特性

- 界面中直接支持多文档编辑，支持设置编辑器中的字体
- 支持复杂的搜索和定位功能
- 增加了 Multi-Physics 模块，支持界面实现多场耦合问题的有限元文件编写，更加符合用户使用逻辑
- 在 AppWizard 功能升级，操作更加简单、人性化

- 开辟了控制台窗口，支持客户端和服务端的程序运行信息直接在 FEPG 主窗口显示

## **2.4. 文档的新特性**

- 产品说明书重新编排，更加面向用户使用，可大大降低用户的学习成本
- 增加了 FEPG 算例集及其文档说明，可以大大提高用户学习效率

## 3. 硬件要求

### 3.1. 服务端

产品模块 硬件配置	SRL.NET	OMP.NET	MPI.NET
CPU	32 位或 64 位体系结构的各种处理器		
RAM	512M	512M	512M
64 位支持	✓	✓	✓
多处理器(多内核)支持	✓	✓	✓
安装所需磁盘空间	100M	120M	130M

### 3.2. 客户端

产品模块 硬件配置	SRL.NET	OMP.NET	MPI.NET
CPU	32 位或 64 位体系结构的各种处理器		
RAM	>=512M	>=512M	>=512M
64 位支持	✓	✓	✓
多处理器(多内核)支持	✓	✓	✓
安装所需磁盘空间	100M	150M	100M

### 3.3. 计算端

产品模块 硬件配置	SRL.NET	OMP.NET	MPI.NET
CPU	32 位或 64 位体系结构的各种处理器		
RAM	>=512M	>=1G	---
64 位支持	✓	✓	✓
多处理器(多内核)支持	支持 PC	支持多内核 PC 或 SMP 并行机	支持 SMP 或 DMP 并行机
网络互连方式	不适用	不适用	- Ethernet - Myrinet - InfiniBand

## 4. 操作系统及支撑环境

### 4.1. 服务端

产品模块	SRL.NET	OMP.NET	MPI.NET
支撑环境			
操作系统	Linux/Unix		
HTTP 服务	Apache		
其他	GCC		

### 4.2. 客户端

产品模块	SRL.NET	OMP.NET	MPI.NET
支撑环境			
操作系统	Linux/Unix/Windows	Linux/Unix/Windows	Linux/Unix/Windows
Fortran 编译器	<b>Windows</b> - Intel Compiler - Compaq Visual Fortran - Powerstation - G77  <b>Linux/Unix</b> - Intel Compiler - G77	<b>Windows</b> - Intel Compiler  <b>Linux/Unix</b> - Intel Compiler	不需要
数学库	不需要	Intel MKL(若客户端与计算端不分离)	不需要

### 4.3. 计算端

产品模块	SRL.NET	OMP.NET	MPI.NET
支撑环境			
操作系统	Linux/Unix/Windows	Linux/Unix/Windows	Linux/Unix

<b>Fortran 编译器</b>	<b>Windows</b> - Intel Compiler - Compaq Visual Fortran - Powerstation - G77	<b>Windows</b> - Intel Compiler	- Intel Compiler - G77
	<b>Linux/Unix</b> - Intel Compiler - G77	<b>Linux/Unix</b> - Intel Compiler	
<b>C/C++ 编译器</b>	不需要	Visual Studio C++ 2003	- Intel Compiler - GCC
<b>MPI</b>	不必需要	不必需要	- Intel MPI - HP MPI - MPICH
<b>OpenMP</b>	不必需要	编译器需支持	不必需要
<b>数学库</b>	不必需要	Intel MKL	火箭并行库

## 5. FEPG.NET V6.0 的绝对优势

FEPG.NET V 6.0 提供了一个能够运行于各种架构的 PC 或并行机上的极经济的 HPC 解决方案,是业内唯一将串行、多线程并行和分布式并行计算完全集于一身的有限元程序开发和计算平台。基于其所作的各种计算能够完全发挥多内核 PC 和并行机所带来的计算能力,可以极其方便地实现部署和扩展。

### 5.1. 最本质的多物理场耦合

多场耦合问题是当前 CAE 领域的热点和难点。飞箭认为,求解多物理场耦合的本质就是求解偏微分方程组。但是由于各个物理场的物理属性的本质差异,一般情况,每个 PDE 所对应的数值求解方法存在着严格的不同。飞箭的多场耦合功能最大限度地发挥了 FEPG 的优势,并在以下几个方面形成了完美耦合

- ☞ 面向偏微分方程编写有限元文件,每个方程对应其唯一的算法
- ☞ 程序统一,计算模拟环境统一
- ☞ 便捷的数据耦合描述
- ☞ 组件程序间有统一的数据接口

### 5.2. 最方便的部署与管理

FEPG.NET V6.0 在部署上延续了 IFEPG 5.2 的架构,采用 C-S,即 Client-Server 架构。

服务端安装 FEPG.NET V6.0 的服务程序,用户解析客户端用户提交的有限元文件并依此生成 Fortran 源代码,借助互连网络返回到客户端。程序对软硬件资源占有率低,安装及维护过程极为简单,管理成本极低。

客户端安装 FEPG.NET V6.0 的客户端程序,包括界面和前后处理系统。用户通过客户端编写有限元文件,向服务端提交命令请求,并接收服务端返回的 Fortran 代码。客户端已

经安装，便可与建立连接的服务器直接通信，无需许可证。管理及维护亦相当容易。

### 5.3. 最创新的应用成果

2007 年，大多数处理器将配备多个内核，但开发人员和用户必须对软件进行测试和调整，才能享受多内核处理器所带来的好处。FEPG.NET V6.0 应时而变，加入了用于开发适应于多内核架构 PC 机及各种架构并行机的模块。目前，由 FEPG.NET V6.0 开发的代码能够适应几乎所有的有限元计算要求和硬件设备。

基于 OpenMP 的并行程序在结构上与通常串行程序无大差别，便于理解和修改，用户可参与能力强，FEPG 的灵活性和扩展性不会受到任何限制。

基于 MPI 的分布式并行程序结构相对复杂，代码级的程序编制对程序员要求高，必须具备数学和计算机的优秀基础。FEPG.NET V6.0 则规避了这些并行有限元计算程序开发的瓶颈问题。用户只需具备应用 FEPG 进行串行计算的基础，便可零时间成本、零难度完成并行代码的研制。

## 6. 结束语

FEPG.NET V6.0 将最新架构计算机的能力、FEPG 的灵活性以及性能优异的编译器和数学库结合在一起，提供了一个低价的有限元 HPC 解决方案。用户在享受快速、高效计算的同时，能够更有力的利用 FEPG 的独有技术实现工程与科学计算领域的创新工作，必将获得更大的收益和价值。

## 7. 附录

### 7.1. OpenMP

OpenMP 起源于 ANSI X3H5 标准，它具有简单、移植性好和可扩展等优点，是共享存储

系统编程的一个工业标准。实际上 OpenMP 并不是一门新的语言，它是对基本语言（如 Fortran77、Fortran90、C、C++ 等）的扩展。OpenMP 规范中定义的制导指令、运行库和环境变量，能够使用户在保证程序的可移植性的前提下，按照标准将已有的串行程序逐步并行化。制导指令是对程序设计语言的扩展，进一步提供了对并行区域、工作共享、同步构造的支持，并且支持数据的共享和私有化。这样，用户对串行程序添加制导指令的过程，就类似于进行显式并行程序设计。运行库和环境变量，使得用户可以调整并行程序的执行环境。OpenMP 的提出，是希望遵循该并行编程模型的并行程序，可以在不同的产商提供的共享存储体系结构间比较容易地移植。实际上，已经有许多硬件和软件供应商提供支持 OpenMP 的编译器，如 DEC、Intel、IBM、HP、Sun、SGI、及 U.S.DOE ASCI program 等，并且包括了 UNIX 和 NT 两种操作系统平台。目前，Fortran77、Fortran90、C、C++ 语言的实现规范已经完成，详细说明可参看 <http://www.openmp.org>。

FEPG.NET V6.0 的 OMP 模块可以生成基于 OpenMP 工业标准的并行程序，但开发和生成过程中并不需要用户了解 OpenMP 的编程方法和规则，完全由 FEPG.NET 自动完成。

## 7.2. MPI

MPI 是一种基于消息传递模型的并行编程接口，目前已经发展成为消息传递模型的代表和事实上的工业标准，而不是一门具体的语言。迄今为止，所有的并行计算机制造商都提供对 MPI 的支持，因而从理论上说任何一个正确的 MPI 程序可以不加修改地在所有并行计算机上运行。MPI 只是一个并行编程语言标准，要编写基于 MPI 的并行程序，还必须借助某一 MPI 具体实现。MPICH 是 Linux 平台下最重要的一种 MPI 实现，是一个与 MPI 规范同步发展的版本。详细说明可参看 <http://www-unix.mcs.anl.gov/mpi/>。

## 7.3. 编译器和数学库

FEPG.NET V6.0 开发的串行程序能够支持所有常见的 Fortran 编译器，并不依赖于任何数学库。

OMP 模块目前只支持 Intel Fortran 编译器，并依赖 Intel 数学核心库(MKL)中的稀疏矩阵解算器 PARDISO。



英特尔® Visual Fortran 编译器支持从 Fortran 2003 选出的新功能和通过 OpenMP\* 进行的并行。它还集成了 Microsoft Visual Studio\* 2005 和 Visual Studio .NET\* 2002 和 2003，可提供扩展的 32 位和 64 位多内核处理器支持。它同康柏 Visual Fortran\* 具有广泛的兼容性，包括语言扩展功能、预先定义的模块和库程序。详见 <http://www.intel.com/software>



英特尔® 数学内核库 (英特尔® MKL) 提供经过高度优化、能够确保线程安全的数学例程，它们专门针对性能要求很高的工程、科学及金融等领域的应用程序而设计。详见 <http://www.intel.com/software>

MPI 模块目前支持各种能够在 Linux/Unix 集群环境下运行的编译器和 MPI 函数库。