DOI：10.7652/xjtuxb201705009

图形处理器上免组装有限元屈曲分析

卞翔，方宗德
（西北工业大学机电学院，710072，西安）

摘要：为了提高大型三维有限元屈曲分析的速度，克服大规模屈曲拓扑优化的计算速度制约，提出了一种基于免组装有限元的线性屈曲分析算法。针对屈曲分析涉及应力刚度矩阵的特殊性，使用了逆迭代法求解特征值问题；利用体素网格的单元一致性，免组装有限元避免了总体刚度矩阵的组装和存储，减少了计算过程中的内存占用，且有利于并行运算；在图形处理器（GPU）上进行稀疏矩阵与向量乘积的运算，利用并行运算进一步加速了有限元的求解速度。算例结果表明，该算法能有效提高大型三维结构屈曲分析的计算速度，与商用软件 Ansys、HyperWorks 相比，计算时间可减少 60%以上，且随着模型自由度的增加，计算速度提高的程度更加显著。

关键词：屈曲分析；免组装有限元；体素化；图形处理器
中图分类号：TH123；O343 **文献标志码**：A **文章编号**：0253-987X(2017)05-0054-06

Assembly-Free Buckling Analysis on Graphics Processing Unit

BIAN Xiang，FANG Zongde
（School of Mechanical Engineering，Northwestern Polytechnical University，Xi'an 710072，China）

Abstract：In order to improve the computational efficiency of large-scale 3D finite element analysis for buckling problem，and to overcome the limitation of computation speed for the large-scale buckling topology optimization，this paper presents a linear buckling analysis algorithm based on the assembly-free finite element method. For the particularity that buckling analysis involves stress stiffness matrix，an inverse iteration method is used to solve eigenvalue problems. In the assembly-free method，the structure is discretized by uniform voxels and there is no need to assemble and store the global stiffness matrix. So the memory footprint is reduced，which is beneficial to parallel computation. The sparse matrix-vector multiplication is performed on GPU (graphics processing unit)，so the parallel computation can further accelerate the speed of finite element analysis. Numerical examples demonstrate that this algorithm can improve the speed of large-scale 3D linear buckling analysis. Compared with the commercial software Ansys and HyperWorks，the computing time of this algorithm can be reduced by more than 60%，and the improvement of the computing speed becomes more obvious with the increase of the model's degree of freedom.

Keywords：buckling analysis；assembly-free method；voxelization；graphics processing unit

屈曲是指当结构受压时偏离原来平衡位置，丧失稳定性的现象。屈曲分析在航空航天、建筑、机械

收稿日期：2016-11-04。 作者简介：卞翔（1988—），男，博士生；方宗德（通信作者），男，教授，博士生导师。 基金项目：国家自然科学基金资助项目（51375384）。
网络出版时间：2017-02-28 网络出版地址：http://www.cnki.net/kcms/detail/61.1069.T.20170228.0809.008.html

等领域有着重要的应用[1]。屈曲分析包括线性和非线性屈曲分析，基于有限元的特征值屈曲分析是基本的线性方法，用于分析结构的临界屈曲载荷和屈曲模态。

Rahmatalla 等使用伴随法进行灵敏度分析，提出以最大化屈曲载荷为目标的拓扑优化模型[2]；Lindgaard 等基于 SIMP(solid isotropic microstructures with penalization)方法以最大化临界屈曲载荷为目标进行拓扑优化[3]。隋允康等基于独立、连续、映射(ICM)方法建立了以结构重量最小为目标、考虑屈曲临界力为约束的连续体拓扑优化模型[4]；Gao 等以最小化柔度为目标，提出屈曲约束的拓扑优化模型[5]。这些研究都只是针对小规模二维问题，虽然二维问题可扩展到三维问题，但三维拓扑优化问题面临着大型三维屈曲分析速度的制约。

Kim 等将薄壁杆件的稳定性控制微分方程组转换成一阶常微分方程组，以其特征值、特征向量为基础构造位移场，建立了刚度阵和应力刚度矩阵并计算了临界荷载[6]；李文雄等构造了精确单元刚度矩阵和应力刚度矩阵，并确定了屈曲载荷及模态[7]。这些方法采用精确单元，无需进行网格细分就可获得精确的屈曲载荷及模态，但对于复杂的三维结构模型需要较多的网格。对于大自由度复杂结构的特征值问题，常采用预处理迭代算法。Arbenz 等提出的预处理迭代算法存在收敛缓慢、或者提前终止迭代而精度低的现象[8]；Golub 等用子空间法对算法进行了改进，提高了算法的稳健性、收敛性，对于大型模态分析非常有效[9]。但是，屈曲分析涉及应力矩阵的特殊性，比常见的模态分析特征值问题更为复杂，这些子空间预处理算法还不能直接应用到屈曲分析问题。

本文使用逆迭代法处理应力刚度矩阵的特殊性，使用体素网格模型，基于网格的一致性，利用免组装有限元来提高有限元分析的速度，该方法收敛速度快且利于并行化计算；同时，基于 CUDA(compute unified device architecture)架构在 GPU 上进行稀疏矩阵与向量乘积运算，使用 GPU 并行计算进一步加速求解速度。

1　有限元屈曲分析

线性屈曲分析是一种常用的计算临界屈曲载荷的方法，广义特征值问题[10]为

$$(\boldsymbol{K}+\lambda_i\boldsymbol{K}_\sigma)\boldsymbol{v}_i=0 \tag{1}$$

式中：$\boldsymbol{K}$ 为稀疏正定的总体刚度矩阵；$\boldsymbol{K}_\sigma$ 为总体应力刚度矩阵；λ_i 为第 i 阶特征值；$\boldsymbol{v}_i$ 为相应的特征向量。最低阶特征值 λ_1 对应的是屈曲发生的临界载荷系数，$\boldsymbol{v}_1$ 是相应的屈曲模态。

使用有限元进行线性屈曲分析包括两个阶段。第 1 阶段是在结构上施加一个单位载荷，使用有限元法求解线弹性问题

$$\boldsymbol{K}\boldsymbol{d}=\boldsymbol{f} \tag{2}$$

式中：$\boldsymbol{d}$ 为位移向量；$\boldsymbol{f}$ 为载荷向量。

在第 2 阶段，基于已求得的位移场 $\boldsymbol{d}$，计算出每个单元的应力张量

$$\boldsymbol{S}^{\mathrm{e}}=\begin{bmatrix}\boldsymbol{\sigma}_x & \boldsymbol{\tau}_{xy} & \boldsymbol{\tau}_{xz}\\ \boldsymbol{\tau}_{xy} & \boldsymbol{\sigma}_y & \boldsymbol{\tau}_{yz}\\ \boldsymbol{\tau}_{xz} & \boldsymbol{\tau}_{yz} & \boldsymbol{\sigma}_z\end{bmatrix} \tag{3}$$

使用这些应力张量计算单元应力刚度矩阵

$$\boldsymbol{K}_\sigma^{\mathrm{e}}=\int\boldsymbol{G}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{S}^{\mathrm{e}} & & \\ & \boldsymbol{S}^{\mathrm{e}} & \\ & & \boldsymbol{S}^{\mathrm{e}}\end{bmatrix}\boldsymbol{G}\mathrm{d}V \tag{4}$$

式中：$\boldsymbol{G}$ 为由形状函数导数构成的矩阵[11]。

求解式(1)，得到特征值 λ_1 和特征向量 $\boldsymbol{v}_1$。临界屈曲载荷为

$$\boldsymbol{L}=\lambda_1\boldsymbol{f} \tag{5}$$

2　基于免组装有限元的屈曲分析

本文基于刚度矩阵无总装的收缩共轭梯度法，采用逆迭代法求解线性屈曲问题。该算法的高效之处在于：体素化网格可以通过编程实现自动的网格划分，提高网格划分的速度；由于体素网格的单元一致性，使用免组装有限元，内存中只需要存储和提取统一的单元刚度矩阵，减少内存占用，在 GPU 上进行并行运算，提高线性方程组的求解速度；使用收缩共轭梯度法加速迭代速度；使用逆迭代求解特征值问题，将特征向量的迭代过程当作静力学问题的等价问题处理。

2.1　体素化网格

体素化是一种简单的有限元网格离散方法，通过统一的体素(六面体单元)对几何模型划分网格。由于单元的形状和尺寸都是一样的，故体素化网格具有稳健性和内存占用空间低的优点，并且可以配合免组装方法，提高有限元分析的速度。

2.2　免组装的收缩共轭梯度法

免组装有限元法最初是 Hughes 等提出的[12]，随着并行算法的发展，这种方法得到了改进[13]。免组装有限元方法的基本原理是单元刚度矩阵不组装成总体刚度矩阵，而是在单元层级进行矩阵向量乘

法。组装然后相乘的过程为

$$\boldsymbol{Kd} = \prod(\boldsymbol{K}_e)\boldsymbol{d} \tag{6}$$

相乘然后组装的过程为

$$\boldsymbol{Kd} = \prod(\boldsymbol{K}_e\boldsymbol{d}_e) \tag{7}$$

共轭梯度法(CG)是求解大型稀疏线性方程的一种迭代算法,迭代过程中涉及的稀疏矩阵向量乘积运算是最主要的计算代价,而大部分的内存占用来自存储和提取刚度矩阵。如果使用统一的单元进行网格划化,所有的单元形状和尺寸相同,单元刚度矩阵相同,因此不需要组装并存储总体刚度矩阵,而只需要储存单个单元刚度矩阵,这将大大减少内存占用,提高矩阵向量乘法运算的速度,从而提高迭代算法的速度。

收缩共轭梯度法(DCG)是一种迭代加速算法[14],此法将有限元网格的点划分成少数的几块,把每块当作刚体处理,构造出收缩空间。将模型的网格分为 200 组,网格分块如图 1 所示。在某块一个点的位移可表示为

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \end{pmatrix} \begin{pmatrix} u_0 \\ v_0 \\ w_0 \\ \theta_x \\ \theta_y \\ \theta_z \end{pmatrix} \tag{8}$$

式中:$(u_0, v_0, w_0, \theta_x, \theta_y, \theta_z)^{\mathrm{T}}$ 为某块 6 个自由度上的刚体运动;(x, y, z)为块中一点相对此块几何中心的坐标。通过所有点的相对坐标,可构造出

$$\boldsymbol{d} = \boldsymbol{W}\boldsymbol{\mu} \tag{9}$$

式中:$\boldsymbol{d}$ 为 $3N$ 个自由度,N 为所有节点个数;$\boldsymbol{W}$ 为收缩矩阵;$\boldsymbol{\mu}$ 为 $6M$ 个与块相关的自由度,M 为分块个数。可以使用收缩空间矩阵进行共轭梯度算法[15],由于 M 远小于 N,所以收缩矩阵的尺寸比刚度矩阵的小,在迭代过程中能提高矩阵向量乘法运算的速度,从而对迭代过程起到加速作用。

图 1 网格分块

2.3 逆迭法

屈曲分析中涉及的广义特征值问题类似于模态分析问题。对于模态分析问题,本可以用 RCG (Rayleigh-Ritz conjugate gradient)求解,但是对于屈曲问题,每个单元应力刚度矩阵依赖于各单元的应力张量,存储每个单元的应力刚度矩阵将会增加内存占用,增加计算时间,因此无法用 RCG 发挥单元一致性的优势。针对这个特殊性,本文采用逆迭代法求解屈曲分析涉及的特征值问题,逆迭代法的基本原理[16]为

$$\boldsymbol{y} = -\boldsymbol{K}^{-1}(\lambda\boldsymbol{K}_\sigma\boldsymbol{v}) \tag{10}$$

此问题可看作静力学问题的等价问题,因此仍然可以使用免组装收缩共轭梯度法求解,大幅降低计算时间。

屈曲问题的逆迭代算法如下:

1. 初始化向量 $\boldsymbol{v}^{(1)}$;
2. 设迭代步数 $i=1$;
3. 计算

$$\lambda^{(i)} = \frac{\boldsymbol{v}^{(i)\mathrm{T}}\boldsymbol{K}\boldsymbol{v}^{(i)}}{\boldsymbol{v}^{(i)\mathrm{T}}\boldsymbol{K}_\sigma\boldsymbol{v}^{(i)}} \tag{11}$$

4. 求解 $\boldsymbol{K}\boldsymbol{v}^{(i+1)} = -\lambda^{(i)}\boldsymbol{K}_\sigma\boldsymbol{v}^{(i)}$,得到 $\boldsymbol{v}^{(i+1)}$;
5. 通过式(11)计算 $\lambda^{(i+1)}$;
6. 计算 $\boldsymbol{g}^{(i)} = \boldsymbol{K}\boldsymbol{v}^{(i+1)} + \lambda^{(i+1)}\boldsymbol{K}_\sigma\boldsymbol{v}^{(i+1)}$;
7. 如果 $\| \boldsymbol{g}^{(i)} \| \leqslant \varepsilon$,结束迭代;否则,增加迭代步数 i,并返回步骤 3。

若特征向量 $\boldsymbol{v}$ 收敛,特征值 λ 可通过式(11)得到。

此算法的核心思想是将步骤 4 的计算当作等效的静力学问题来处理,使用免组装收缩共轭梯度算法求解,可简化程序的实现过程。因为数值误差已经预先在步骤 4 线性问题的求解中排除了,所以算法使特征向量收敛所需的迭代步数相比 RCG 算法大幅减少。

2.4 基于 GPU 的并行运算

用于应力计算的 DCG 算法和用于特征值求解的逆迭代算法中都涉及稀疏矩阵与向量乘积(SpMV),其计算效率对整个算法的速度有重要作用。本文使用 CUDA 在 GPU 上进行 SpMV 的并行运算。GPU 执行的最小单位是线程,根据式(7),应将每个单元分配给线程运算,然后按单元更新结果,但在并行运算中这会造成竞态条件,因为多个进程会同时存取连接不同单元的一个点,所以改为将每个点分配给线程。体素网格中每个点最多连接 8 个体素单元、27 个点,提取这些相邻单元的刚度矩阵系数和位移信息进行计算,避免了并行运算中的竞态条件。

并行运算 $\boldsymbol{y}=\boldsymbol{K}\boldsymbol{x}$ 的算法如下:

1. 分配线程到每个点上;
2. 设 $y(3n:3n+2)=0$,n 为线程的编号;
3. for 与该点相连的单元 e,读取单元刚度矩阵 $\boldsymbol{K}_e$ 和点的信息 $\boldsymbol{x}_e$,计算 $\boldsymbol{y}_e=\boldsymbol{K}_e\boldsymbol{x}_e$,累加 $\boldsymbol{y}_e$ 到 $y(3n:3n+2)$,end;
4. 同步线程。

由于使用了一致的体素单元,所有线程运行时共享相同的单元刚度矩阵,内存读取一次单元刚度矩阵就足够了,这有利于在 GPU 上的并行运算。

3　算　例

本文算法是用 C 语言结合 CUDA 平台编程来实现的,通过算例与商用软件 Ansys、HyperWorks 进行对比,弹性模量均为 2.1×10^{11} Pa。

3.1　长方形平板的屈曲

算例 1　考虑长方形平板,尺寸为 1 000 mm×100 mm×10 mm,板的一段固定,另一端施加单位压缩力 1 N,使用经典材料力学中的 Euler 公式求得临界屈曲载荷系数为 4 313.6 N。

本文方法分析的结果和 Ansys、HyperWorks 得到的临界屈曲载荷结果对比如图 2 所示。3 种方法使用相同数量的自由度,随着网格数量的增加,计算结果都趋于稳定并逐渐接近理论计算值,Ansys 的结果为 4 347.0 N,HyperWorks 的结果为 4 366.1 N,本文方法的结果为 4 343.8 N。

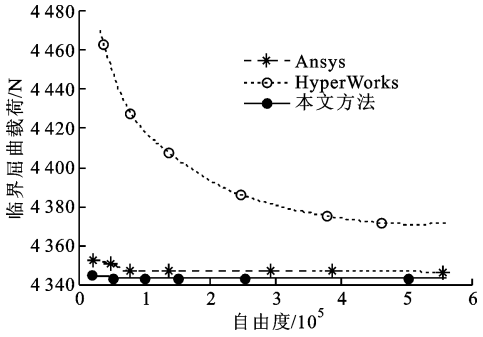


图 2　算例 1 的临界屈曲载荷

本文方法的优势是计算速度快。图 3 给出了本文方法和 Ansys、HyperWorks 的计算时间对比。Ansys、HyperWorks 线性屈曲分析的默认算法都是 Lanczos 法,Lanczos 法是求解特征值问题的一种直接算法,该算法的优点是稳定性高,但缺点是对于大自由度问题,需要较大的内存占用和较高的计算时间。由图可知,Ansys、HyperWorks 的计算时间和变化趋势比较一致,随着网格数量增加,所需的计算时间快速增长。这是因为随着自由度的增加,所需内存占用急剧增长。本文方法发挥了免组装有限元对于求解大规模问题的优势,计算时间显著减少。

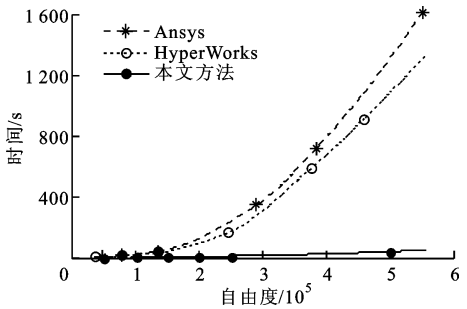


图 3　算例 1 的计算时间

3.2　细长圆杆的屈曲

算例 2　考虑一个细长圆杆,截面半径为 10 mm,长度为 1 m,杆的一段固定,另一端施加单位压缩力 1 N,使用 Euler 公式求得临界屈曲载荷系数为4 063 N。

使用本文算法的结果和商用软件 Ansys、HyperWorks 得到的结果对比如图 4 所示,3 种算法的结果分别为 4 203、4 072 和 4 042 N,Ansys、HyperWorks 的计算结果更加接近公式计算值。本文方法存在误差的原因在于体素化网格,由于使用了固定的六面体单元,对于模型的圆形表面无法完全精确表示,而对于长方形平板,由于体素网格能够精确建模,所以两种方法的结果误差很小。

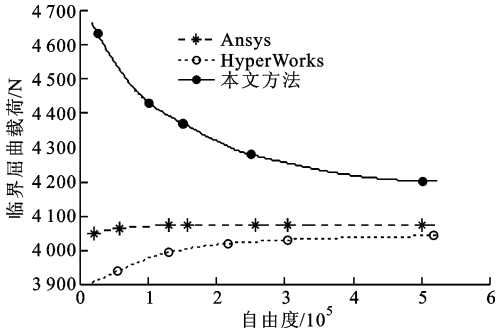


图 4　算例 2 的临界屈曲载荷

事实上,线性屈曲分析本身并不十分精确,在实际应用中通常用于非线性屈曲分析的初步评估,其主要目的并不是精确计算结构失稳的临界值,而是预测临界失稳力的大致所在。另外,考虑到本文的研究目的是为了提高屈曲拓扑优化的基础,对于拓扑优化,各单元的相对灵敏度比精度更为重要。图 5 所示是 3 种方法在不同网格密度下的时间对比,计算时间的趋势和算例 1 中的一致。如果能够接受一定的误差,本文方法的计算速度有显著优势。

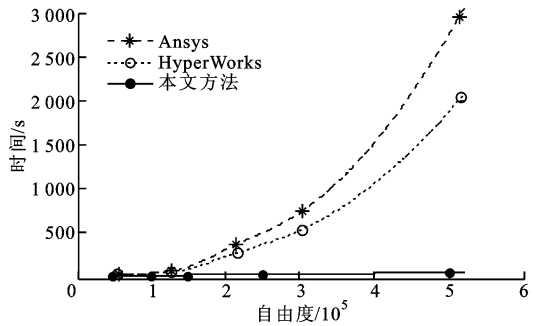
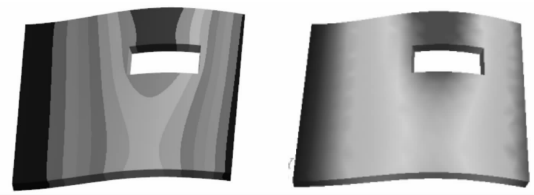


图 5 算例 2 的计算时间

3.3 带孔平板的屈曲

算例 3 考虑不规则的结构及约束条件。一个带孔的方形平板，一边固定，另一边限制 Z 向自由度，并施加单位载荷 1 N，屈曲模态如图 6 所示。图 7 为临界屈曲载荷的对比，可知随着单元数量增多，网格质量得到提高，3 种算法的计算结果都趋于稳定且误差很小，Ansys 为 398 kN，HperWorks 为 393 kN，本文方法为 394 kN。由于计算时间与有限元模型的自由度有关，而与具体结构无关，所以算例 3 的计算时间趋势和前 2 个算例一致，都是随着模型自由度的增加，本文方法能体现出明显的计算速度优势。



(a)Ansys (b)本文方法

图 6 屈曲模态

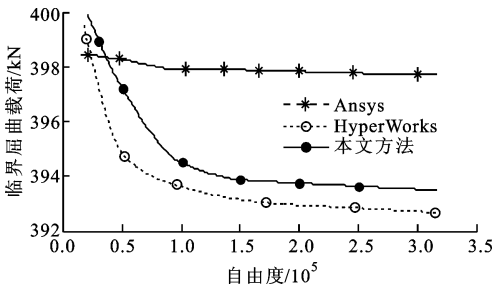


图 7 算例 3 的临界屈曲载荷

本文方法在不同分组数下的计算时间如表 1 所示，自由度为 2×10^5。由表 1 可知：在未分组的情况下，计算时间为 253 s，而通过对网格进行分块凝聚后加速作用明显，其中分为 200 组的计算时间仅为未分组的 5%；随着分组数量的增加，加速程度逐渐提高，但当分组数量增加到一定程度后，加速效果会开始降低，如何找到最佳分组数量还需要进一步研究。分组过程通过编程在体素网格的生成过程中完成，几乎不影响网格划分时间。图 8 给出了不同分组情况下的收敛性，通过分块加速后，有限元求解所需的迭代次数大幅降低，所以计算时间大幅减少。

表 1 不同分组数对应的本文算法计算时间

分组数	网格生成时间/s	计算时间/s
0	2.38	253
10	2.38	75
100	2.38	17
200	2.41	13
300	2.39	14
500	2.39	15

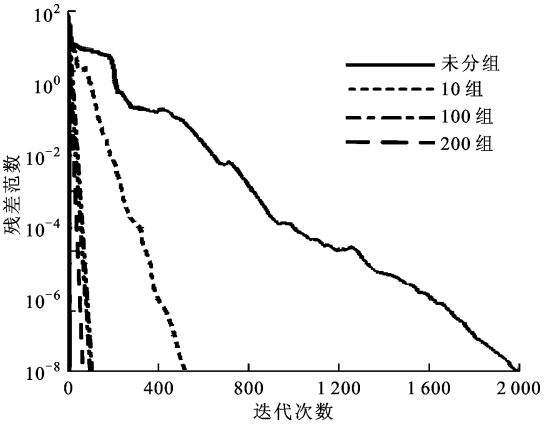


图 8 不同分组情况下的收敛性

3.4 工程实例

屈曲是弧形连杆失效形式之一，设计时需要考虑屈曲问题。图 9 为汽车多连杆悬架中的一个弧形连杆，其上沿的宽度为 20 mm，厚度为 7 mm，下肋板的厚度为 5 mm。使用本文方法进行屈曲分析，固定左侧圆孔，在右侧圆孔施加一个单元压力。临界屈曲载荷为 14 500 N，网格模型的自由度为 167 000，计算时间为 24 s。在相同自由度下，Ansys 的计算时间为 112 s，临界屈曲载荷为 14 730 N。当连杆承受屈曲载荷时，最大应力为 419 MPa，低于材料屈服强度 620 MPa，这意味着该弧形连杆会由于屈曲而失效，因为屈服现象会在材料到达屈服极限前

(a)连杆模型 (b)屈曲模态

图 9 屈曲分析

先发生，所以需要对此连杆进行改进设计。

4　结　论

本文所提基于免组装有限元的线性屈曲分析算法，将免组装收缩共轭梯度法用于求解静力学线性方程组及特征值问题的迭代过程中，利用 GPU 并行运算，可有效提高大型三维结构屈曲分析的计算速度。由于体素化网格的局限性，对于一些几何细节无法精确表示，导致屈曲分析产生一定误差，但与降低的计算时间相比，精度下降程度很小，产生的误差可以接受，而且线性屈曲分析本身不很精确，在实际应用中，一般不直接用于取得工程问题的精确解，而是用于非线性屈曲分析之前确定屈曲发生的大致载荷，所以快速给出参考值更为重要。此外，本文的目的是为大规模屈曲拓扑优化提供理论基础，拓扑优化是设计初期的概念设计，计算速度的重要性大于计算精度。

参考文献：

[1] 李瑞雄，陈务军，付功义. 透镜式薄壁 CFRP 管空间伸展臂轴压屈曲分析及试验 [J]. 宇航学报，2012，33(8)：1164-1170.
LI Ruixiong，CHEN Wujun，FU Gongyi. Buckling analysis and experiment of lenticular CFRP thin-walled tube space boom under axial compression [J]. Journal of Astronautics，2012，33(8)：1164-1170.

[2] RAHMATALLA S，SWAN C C. Form finding of sparse structures with continuum topology optimization [J]. Journal of Structural Engineering，2003，129(12)：1707-1716.

[3] LINDGAARD E，DAHL J. On compliance and buckling objective functions in topology optimization of snap-through problems [J]. Structural & Multidisciplinary Optimization，2013，47(3)：409-421.

[4] 隋允康，边炳传. 屈曲与应力约束下连续体结构的拓扑优化 [J]. 工程力学，2008，25(8)：6-12.
SUI Yunkang，BIAN Bingchuan. Topology optimization of continuum structures under buckling and stress constraints [J]. Engineering Mechanics，2008，25(8)：6-12.

[5] GAO X，MA H. Topology optimization of continuum structures under buckling constraints [J]. Computers & Structures，2015，157：142-152.

[6] KIM N I，DONG K S，KIM M Y. Improved flexural-torsional stability analysis of thin-walled composite beam and exact stiffness matrix [J]. International Journal of Mechanical Sciences，2007，49(8)：950-969.

[7] 李文雄，马海涛，高兴军. 开口薄壁杆件结构稳定分析的精确单元和两步求解算法 [J]. 计算力学学报，2012，29(3)：405-411.
LI Wenxiong，MA Haitao，GAO Xingjun. An exact finite element and a two-step algorithm for stability analysis of frame structures with open thin-walled sections [J]. Chinese Journal of Computational Mechanics，2012，29(3)：405-411.

[8] ARBENZ P，HETMANIUK U L，LEHOUCQ R B. A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods [J]. International Journal for Numerical Methods in Engineering，2005，64(2)：204-236.

[9] GOLUB G H，QIANG Y E. An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems [J]. SIAM Journal on Scientific Computing，2002，24(1)：313-334.

[10] YADAV P，KRISHNAN S. Assembly-free large-scale modal analysis on the graphics-programmable unit [J]. Journal of Computing and Information Science in Engineering，2013，13(1)：1-7.

[11] DÜSTER A，PARVIZIAN J，YANG Z. The finite cell method for three-dimensional problems of solid mechanics [J]. Computer Methods in Applied Mechanics and Engineering，2008，197：3768-3782.

[12] HUGHES T J R，LEVIT I，WINGET J. An element-by-element solution algorithm for problems of structural and solid mechanics [J]. Computer Methods in Applied Mechanics and Engineering，1983，36(2)：241-254.

[13] YADAV P，SURESH K. Large scale finite element analysis via assembly-free deflated conjugate gradient [J]. Journal of Computing and Information Science in Engineering，2014，14(4)：041008.

[14] AUBRY R，MUT F，DEY S. Deflated preconditioned conjugate gradient solvers for linear elasticity [J]. International Journal for Numerical Methods in Engineering，2011，88(11)：1112-1127.

[15] SAAD Y，YEUNG M，ERHEL J. A deflated version of the conjugate gradient algorithm [J]. SIAM Journal on Scientific Computing，1998，21(5)：1909-1926.

[16] IPSEN I C F. Computing an eigenvector with inverse iteration [J]. SIAM Review，1997，39(2)：254-291.

（编辑　赵炜　葛赵青）