

应用 GPU 求解的实时子结构试验架构与性能验证

董晓辉¹, 唐贞云^{1,2}, 李振宝¹, 杜修力¹

(1.北京工业大学城市与工程安全减灾教育部重点试验室,北京 100124;

2.河北省地震灾害防御与风险评价重点实验室,河北 三河 065201)

摘要: 使用图形处理器(GPU)代替传统中央处理器(CPU)作为数值求解硬件,建立基于LABVIEW-MATLAB-GPU的实时子结构试验架构。以土-结相互作用系统为载体,通过数值仿真与试验对该架构的性能进行验证。试验与仿真结果表明,本文方法将实时子结构试验中数值子结构求解自由度提高到27000,提升了数值模型求解规模,拓展了实时子结构试验应用范围。

关键词: 实时子结构试验; 图形处理器; 数值积分算法; 振动台; 有限元模拟

中图分类号: TU311.3; TU352.1

文献标志码: A

文章编号: 1004-4523(2022)01-0064-08

DOI: 10.16385/j.cnki.issn.1004-4523.2022.01.007

引言

实时子结构试验^[1]是一种结合物理试验与数值计算的试验方法,该方法将需要重点研究及难以建模的部分进行物理试验,其他部分采用数值建模的方式进行仿真,两部分之间数据实时交互传输,使得对大型复杂结构进行大尺寸试验成为可能^[2]。在实时子结构试验中,为了保证求解实时性,数值子结构求解通常采用诸如中心差分法等求解效率高的显式积分算法,显式积分算法的缺点是条件稳定性,为了保证求解稳定,积分步长(Δt)越小越好^[3]。过小的积分步长限制了数值子结构计算规模,无法满足实际试验研究需求^[4]。为了解决上述问题,可行的手段主要包括:改进数值求解算法^[5]和提高数值求解效率^[6]。

在算法研究方面,Chang等^[7]、Nakashima等^[8]分别研究了无条件稳定的拟动力显式算法,此方法在实时子结构试验中为隐式方法。Wu等^[9]将上述方法扩展成为了适于实时子结构试验的无条件显式方法。基于既有积分方法,Nakashima等^[10]提出将数值求解分为模型动力分析(RAT)与信号处理生成(SGT)两部分,实现了 $\Delta t=330$ ms条件下10自由度或 $\Delta t=500$ ms条件下12自由度结构实时子结构试验。Cheng等^[11]结合MATLAB软件开发“Hy-

bridFEM”程序,将有限元与实时子结构试验技术相结合,实现了 $\Delta t=10$ ms条件下122自由度结构实时混合试验。Chae等^[12]使用Hybrid-FEM技术开展了 $\Delta t=10$ ms条件下514自由度结构实时混合试验。Saouma等^[13]开发了在实时硬件上运行的Mercury软件,完成了 $\Delta t=10$ ms条件下405自由度的高度非线性模型实时混合试验。Zhu等^[4]实现了 $\Delta t=20$ ms条件下1240自由度结构实时混合试验。综上所述,算法的研究及有限元技术的发展,提升了数值子结构实时求解能力,但目前的求解规模仍在2000以内。

传统的数值子结构计算是基于电脑的中央处理器(CPU)运算,由于硬件架构不同,在大规模数值计算时使用图形处理器(GPU)更有优势。在土木工程领域,GPU技术也得到应用。Durand等^[14]基于GPU并行使用离散元的方法计算模拟岩石与混凝土接触问题,相比CPU,计算速度提高了30倍。解琳琳等^[15]基于GPU对城市地震灾害模拟,对建筑结构非线性分析,相比CPU,计算速度提升了39倍。综上所述,GPU在土木工程领域具有很好的发展前景。这为通过改善硬件能力来提高实时子结构试验系统中数值求解效率创造了条件。基于此,本文提出了基于GPU计算的实时子结构试验系统架构,分别从数值仿真与实时子结构试验两方面讨论了该试验架构的可能性与实际性能。

收稿日期: 2021-01-18; **修订日期:** 2021-03-29

基金项目: 国家自然科学基金面上项目(51978016);河北省地震灾害防御与风险评价重点实验室开放基金资助项目(FZ213104)。

1 试验系统架构

在基于 GPU 求解的实时子结构试验系统中,用 GPU 代替传统 CPU 作为数值子结构求解硬件。为建立基于 GPU 求解的试验系统,有两个问题需要解决:其一,如何实现 GPU 求解的数值子结构与试验子结构之间信号实时交互;其二,如何使用 GPU 求解大规模的数值子结构模型。

1.1 试验系统架构组成

基于 GPU 求解的实时子结构试验系统包括如图 1 所示的三个部分,分别是数值子结构、信号传输

和试验子结构。数值子结构求解环节基于 GPU 硬件实现对数值子结构模型动力响应的实时计算。快速准确的数值模型动力求解是实时子结构试验的必要条件,因此要满足 GPU 数值子结构模型求解实时性和求解过程稳定的要求。解决 GPU 求解实时性的方法可分为软件实时和硬件实时两种方法。软件实时方法是通过 LABVIEW 等软件的定时循环结构控制每一步动力求解所需时间固定,或使用 SIMULINK 仿真软件环境控制数值求解的时间。硬件实时方法是在例如 dSPACE, Speedgoat 等硬件实时系统中编写数值模型动力分析程序,实时性高,数值求解高效稳定。

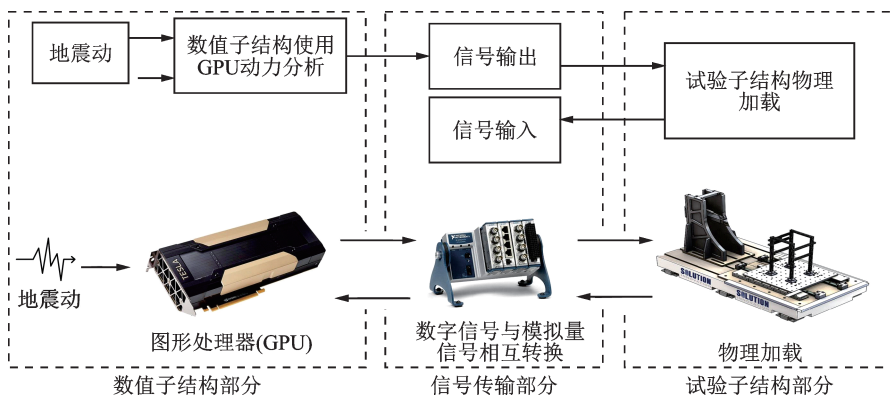


图 1 基于 GPU 的实时子结构试验系统架构

Fig. 1 GPU-based framework for real time hybrid testing

信号传输部分的作用在于确保 GPU 求解的数值子结构与试验子结构之间数据实时通讯。两部分之间可通过数字信号或电信号模拟量通讯。数字信号通讯的方法可选用共享内存卡、TCP/IP 或 UDP 等方法实现两部分之间数字信号传输,此方法抗干扰能力较强,对实验环境要求较低;使用电信号模拟量通讯须将数值求解计算机的发出数字信号转换为电信号模拟量传输到试验子结构部分。此方法易于操作,但易受到试验环境干扰的影响。

试验子结构部分对试验试件物理加载,其功能与传统基于 CPU 的混合试验方式相同。实时子结构试验时,信号传输部分将数值求解所得界面响应传输给加载系统控制器,而后控制加载系统将界面响应施加给物理子结构。通过传感器测得试验子结构对数值子结构的反作用响应,并通过信号传输部分传输到数值子结构。

1.2 GPU 求解方法

目前具有 GPU 求解功能的软件有 ABAQUS, OpenSEES, ANSYS 等有限元软件以及 MATLAB 等数学计算软件。实现基于 GPU 求解数值模型,还

可使用 Python, C++ 等编程语言编写动力分析程序。使用例如 ABAQUS 有限元软件时,需要对试验模型整体建模,在作业设置中设置 GPU 并行计算。试验时将整体试验模型的一部分作为试验子结构,利用软件提供的接口程序与 OpenFresco 等软件交互,实现与物理试验设备的交互^[16-17]。使用 MATLAB 软件时可利用 Parallel Computing Toolbox (PCT) 工具箱, PCT 工具箱支持多核 CPU 及 GPU 并行计算功能,在大型矩阵运算时使用 GPU 并行能显著提升计算效率。在 MATLAB 软件中编写动力分析程序,利用例如 LABVIEW, dSPACE 等即可与物理试验系统数据交互。使用 C++ 等语言编写程序时,可利用 NVIDIA 公司推出的 CUDA 并行计算架构,编写动力分析程序,并添加数据通讯接口功能。其中在大规模矩阵计算时利用 CUDA 库函数实现 GPU 并行求解,可显著提升求解效率。

2 基于 MATLAB 的 GPU 求解性能

如前述可知,可采用 GPU 进行动力求解的软件较多,本文选择 MATLAB 作为数值求解工具验证

基于GPU的实时子结构试验架构的可行性。应用MATLAB软件的PCT工具箱,调用GPU代替CPU进行数值子结构动力分析。使用GPU与CPU对相同模型进行动力时程分析,验证MATLAB软件基于GPU求解的性能。

2.1 实施方案

不同于商用有限元软件具有前处理功能,采用MATLAB进行动力求解前需要建立数值子结构动力方程,数值子结构的刚度和质量矩阵往往借助商用有限元软件的前处理功能得到。在MATLAB软件中使用GPU求解时,需将刚度、质量及阻尼矩阵等数据使用gpuArray函数初始化为GPU格式,同时将数据保存于GPU显存中。使用GPU对数值子结构动力分析时,求解过程的程序与使用CPU计算相同。数值模型动力分析完成后,将数值子结构与物理子结构接触点的交互值传输到信号传输部分,需要使用gather函数将交互值转化为普通格式的数据。使用MATLAB-GPU进行实时子结构试验时数值子结构部分流程图如图2所示。图2中左侧部分为在CPU硬件上的工作,右侧部分为GPU硬件上的工作,主要计算求解在GPU硬件上进行。

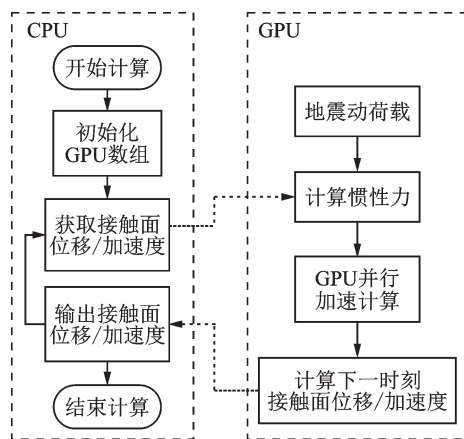


图2 基于GPU的数值子结构求解流程

Fig. 2 Calculation process of GPU-based numerical solution

为了快速建立数值子结构动力方程,本文使用ABAQUS有限元软件完成数值子结构前处理。通过改变模型网格间距调整模型自由度数,及数值子结构求解规模。完成前处理后,在模型参数文件中添加模型参数矩阵输出代码,重新提交作业后即可得到模型的质量、刚度矩阵。从ABAQUS有限元软件中仅能提取质量及刚度参数,模型的阻尼采用如下式所示的瑞利阻尼,通过刚度和质量矩阵构建阻尼矩阵:

$$C = a_0 M + a_1 K \quad (1)$$

式中 C 为阻尼矩阵, M 为质量矩阵, K 为刚度矩阵, a_0 和 a_1 是两个比例参数,根据前两阶模态频率确定。

2.2 仿真参数

为验证本文采用的基于MATLAB-GPU求解数值模型方案的可行性,采用如图3所示的三维土-结相互作用系统作为算例,其中上部结构作为物理子结构,土体作为数值子结构。本文重点研究GPU对数值求解的加速效果,选用求解效率高的中心差分法作为动力分析算法。为满足中心差分法收敛条件,对数值子结构土体模型参数进行调整。数值模型尺寸为 $30\text{ m} \times 30\text{ m} \times 15\text{ m}$,密度为 $1 \times 10^4\text{ kg/m}^3$,弹性模量为 $2.11 \times 10^8\text{ Pa}$,泊松比为 0.33,阻尼比为 0.05,通过对模型四周与底面节点设置弹簧及阻尼器模拟远场土体边界条件。边界节点法向弹簧刚度为 20000 N/m ,法向阻尼器阻尼值为 $1.437 \times 10^6\text{ N/(m/s)}$,切向弹簧刚度为 10000 N/m ,切向阻尼器阻尼值为 $9.45 \times 10^6\text{ N/(m/s)}$ 。在ABAQUS软件中对土体有限元模型划分网格,每个节点有3个自由度,通过改变网格间距的大小控制数值子结构模型自由度数,即数值求解计算量大小。本节的目的在于讨论GPU的数值求解性能,仅对数值子结构进行动力分析。模型荷载为El-Centro波,加速度时程幅值为 $0.48g$,结构动力分析方法为中心差分法。

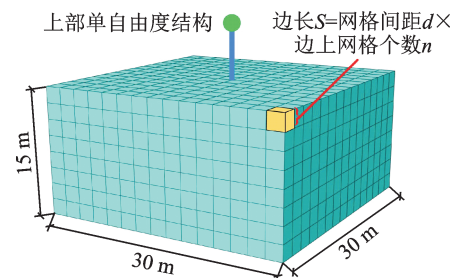


图3 土-结相互作用系统

Fig. 3 Soil-structure interaction system

为了对比基于MATLAB的GPU和CPU求解性能,分别使用装有消费级GPU的笔记本电脑(PC)和装有专业计算级GPU的服务器对比,PC与服务器硬件配置参数如表1所示。

2.3 仿真结果

为了对比CPU与GPU计算效率,需要在计算过程中记录各自的耗时情况。试验中在动力分析算法开始部分添加函数tic,结束部分添加函数toc,记录求解所用的时间。仿真中数值模型自由度分别取3888自由度、6591自由度及27000自由度。使用PC及服务器计算,对比分别使用GPU及CPU求解时

表 1 仿真系统配置参数

Tab. 1 The configuration parameters of simulation system

| 系统配置名 | PC | 服务器 |
|--------|-----------------------|-----------------------|
| CPU 型号 | Intel i5 8300H | Intel Xeon E5-2690 V4 |
| CPU 性能 | 4 核 2.3 GHz | 14 核 2.6 GHz |
| GPU 型号 | NVIDIA GTX 1050 | NVIDIA Tesla V100 |
| GPU 性能 | 3.882 GFLOPS | 10.6 TFLOPS |
| 内存 | 8 GB | 32 GB |
| 操作系统 | Windows 10 64 位 | |
| 软件版本 | MATLAB 2020,CUDA 10.2 | |

每一步长的平均用时,对比结果如表 2 所示。表中 SR (Speedup Ratio) 为加速比,加速比计算公式如下:

$$SR = \frac{T_{CPU}}{T_{GPU}} \quad (2)$$

式中 T_{CPU} 为使用 CPU 求解每一积分步长平均所用时间; T_{GPU} 为使用 GPU 求解每一积分步长平均所用时间。

表 2 求解时间对比(单位:ms)

Tab. 2 Comparisons of solving times (Unit: ms)

| 模型自由度 | PC | | | 服务器 | | |
|-------|-----------|-----------|-----|-----------|-----------|-----|
| | T_{CPU} | T_{GPU} | SR | T_{CPU} | T_{GPU} | SR |
| 3888 | 20 | 10 | 2 | 17 | 2 | 8.5 |
| 6591 | 68 | 21 | 3.4 | 48 | 4 | 12 |
| 27000 | | | | 936 | 17 | 55 |

分析表 2 中的时间对比,当模型自由度为 3888 时,使用 PC 与服务器 CPU 求解所用时间近似,说明在较小计算量运算时,无法发挥服务器的性能。当使用 GPU 求解时,服务器的 GPU 求解速度远超 PC 的 GPU 求解速度,且此时相比服务器 CPU 求解达到 8.5 倍的加速效果。当模型自由度为 6591 时,使用 PC 的 GPU 求解达到 MATLAB 软件求解上限。此时 CPU 计算一个步长需要 68 ms,而 GPU 只需要 21 ms, GPU 的求解能力更适于实时子结构试验需求。

使用服务器 GPU 求解时, MATLAB 软件求解上限自由度为 27000。此时每个步长所需计算时长为 17 ms,说明使用本文配置服务器的 GPU 基于 MATLAB 软件最高可实现 27000 自由度数值模型的实时子结构试验。此时服务器 GPU 相比服务器 CPU 求解可达到 55 倍的加速效果,使用 GPU 进行数值模型求解的优势显著。且此时使用服务器 CPU 求解时需要 936 ms,远不能满足实时子结构试

验的实时性要求。

为了对比使用 CPU 与 GPU 的求解精度,分别使用 CPU 与 GPU 计算,数值模型自由度为 3888 自由度,中心差分法积分步长分别取为 1, 5, 20 ms,对比三个时间步长条件下使用 CPU 与 GPU 求解的数值模型顶部中心点加速度时程响应,响应结果对比如图 4 所示。图 4(a) 为 $\Delta t=1$ ms 使用 CPU 与 GPU 求解时时程曲线对比,可见 CPU 与 GPU 求解结果一致。为进一步研究 GPU 求解精度,图 4(b) 和 (d) 为 $\Delta t=1, 5, 20$ ms,将 GPU 求解位移时程 D_{GPU} 作为 X 轴, CPU 求解位移时程 D_{CPU} 作为 Y 轴绘图。从图中可见均为直线,说明 GPU 求解结果与 CPU 具有同样的精度,可以使用 GPU 代替 CPU 对数值模型进行动力分析,提高求解效率。

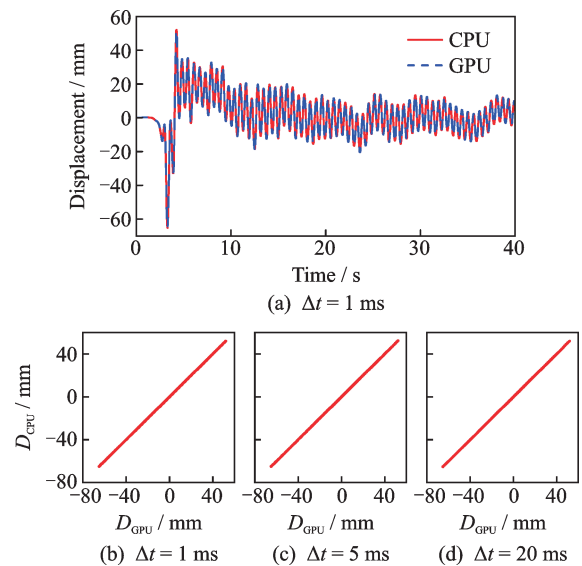


图 4 CPU 与 GPU 求解试验子结构位移时程对比

Fig. 4 Displacement time history of physical substructure based on CPU and GPU

3 试验实现与验证

将基于 MATLAB 软件 GPU 求解数值模型应用于实时子结构试验中,设计基于 GPU 的实时子结构试验系统,并验证试验系统可行性。本文设计试验系统数值子结构部分由 ABAQUS 软件前处理并提取模型参数,在 LABVIEW 软件中调用 MATLAB 软件的工具箱,实现基于 GPU 的数值模型动力分析。LABVIEW 软件保证数值子结构求解的时间固定,同时完成数值与物理子结构之间的数据传输。根据 MATLAB 软件数值求解单步长平均用时及数值积分算法的积分步长,改变 LABVIEW 软件的定时循环结构周期,以保证数值求解的时间固定。

试验加载系统采用振动台,对应的实时试验系

统架构如图 5 所示, 试验子结构对数值子结构的反力与地震动荷载作为数值子结构的外荷载。基于 GPU 的数值子结构求解完成后将数值子结构与试验子结构接触点的位移或加速度信号通过信号传输部分传输到试验子结构。本文试验的目的在于验证基于 GPU 的实时子结构试验实现过程的可行性, 为了避免对研究中物理子结构模型建模误差的影响, 此处的物理子结构也通过数值建模实现, 只有振动台为真实的, 物理子结构通过在 dSPACE 建模仿真。测量振动台台面的位移及加速度, 将信号传输到试验子结构仿真部分。试验子结构部分采用

dSPACE 实时仿真环境进行仿真, 计算动力响应并通过信号传输部分传输到数值子结构部分, 由此构成基于 GPU 求解的实时子结构试验系统。在本文试验系统中数值子结构求解部分与表 1 中服务器配置相同, 其他各部分配置情况如表 3 所示。

表 3 试验系统配置参数

Tab. 3 Testing system configuration parameters

| 系统配置名 | 配置参数 |
|--------|-------------------------------------|
| 信号传输部分 | NICompactDAQ-9174, NI-9201, NI-9263 |
| 实时仿真部分 | dSPACE MicroLabBOX-1202 |

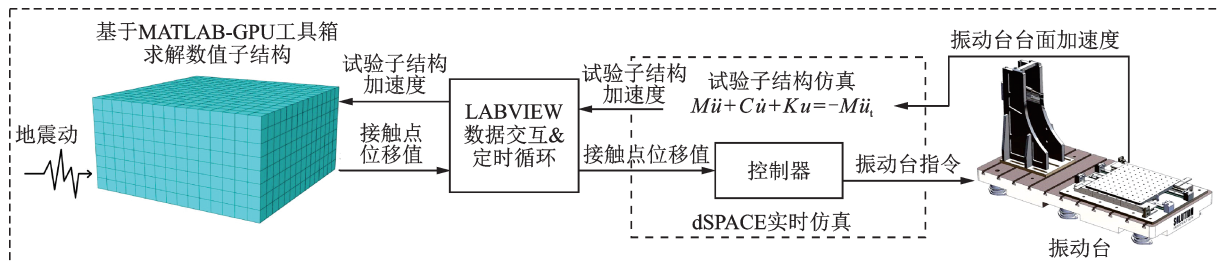


图 5 验证试验实现过程

Fig. 5 Implementation process of verification testing

3.1 试验参数选取

数值子结构模型与第 2.2 节仿真模型相同, 试验子结构选择为水平方向单自由度模型, 位于数值子结构顶部中心与数值子结构之间锚固连接。试验子结构质量 $m=7240$ kg, 刚度 $k=753250$ N/m, 阻尼 $c=440$ N/(m/s)。在振动台台面布置加速度传感器及位移传感器。将白噪声信号输入振动台, 对振动台系统识别, 使用 4 阶传递函数辨识振动台特性, 传递函数如下所示:

$$G_{st} = 1.023 \times 10^9 / (s^4 + 427.5s^3 + 1.481 \times 10^5 s^2 + 3.041 \times 10^7 s + 1.017 \times 10^9) \quad (3)$$

振动台特性与辨识得到的传递函数幅值相位对比如图 6 所示。振动台系统信号在 0~2 Hz 之间时幅值和相位误差都较小, 振动台信号在此频率区间控制较为精准。超过 2 Hz 后, 振动台幅值及相位差随频率增加而增大, 因此需要在信号输入振动台之前添加外环控制器, 对振动台动力特性进行补偿。基于式 (3) 所示振动台动力模型, 采用 Tang 等^[18]提出的 FSCS 补偿控制器进行控制。

本试验系统基于 Windows 系统, Windows 系统是非实时系统, 由于系统实时性不稳定问题, 及 LABVIEW 软件调用 MATLAB 的方法有通讯延迟, 由此造成的信号传输有近 3 ms 的延迟。即数值子结构动力分析完成后需要增加 3 ms 才能发送信

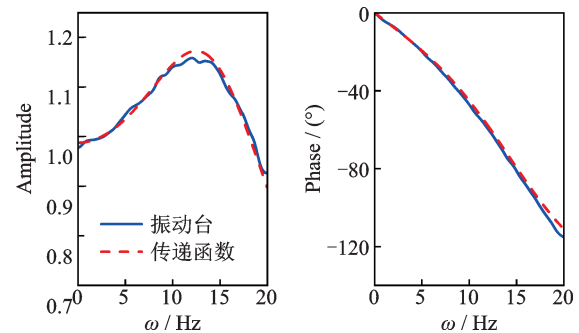


图 6 振动台传递函数

Fig. 6 Transfer function of shaking table

号到信号传输部分, 且在此 3 ms 期间 MATLAB 脚本停止计算。因此试验中最小时间步长如下所示:

$$\Delta t = t_{\text{solve}} + 3 \quad (4)$$

式中 Δt 为中心差分法积分步长, t_{solve} 为数值模型求解所用时间, 由此算法积分步长最小取为 4 ms。步长太大对数值积分算法精度的影响较大, 综合考虑将本试验中最大时间步长取为 20 ms。

3.2 试验结果

影响数值求解精度与效率的因素主要包括硬件性能、积分步长、自由度数和浮点数精度。本节对采用 GPU 和 CPU 求解时, 不同积分步长、不同自由度和不同浮点数精度的试验性能进行了对比。表 4 给出了步长为 4 和 20 ms 时使用 GPU/CPU、双精度/单精度时数值子结构模型的最大自由度数。

表 4 CPU 与 GPU 求解性能对比

Tab. 4 Performance comparison of numerical solution by GPU and CPU

| 工况 | $\Delta t/\text{ms}$ | 求解硬件 | 精度 | 自由度数 |
|----|----------------------|------|-----|-------|
| 1 | 4 | GPU | 双精度 | 1500 |
| 2 | 4 | GPU | 单精度 | 3168 |
| 3 | 4 | CPU | 双精度 | 1080 |
| 4 | 4 | CPU | 单精度 | 1500 |
| 5 | 20 | GPU | 双精度 | 18876 |
| 6 | 20 | GPU | 单精度 | 27000 |
| 7 | 20 | CPU | 双精度 | 2904 |
| 8 | 20 | CPU | 单精度 | 3888 |
| 9 | 5 | GPU | 单精度 | 3888 |
| 10 | 20 | GPU | 单精度 | 3888 |

将子结构试验结果与 MATLAB 整体结构计算结果对比,表 4 中 $\Delta t=4 \text{ ms}$ 、使用 GPU 求解、模型精度为双精度条件下振动台子结构试验结果与整体仿真计算试验子结构顶部位移时程对比如图 7 所示。图 7 中子结构试验与整体仿真结果吻合,

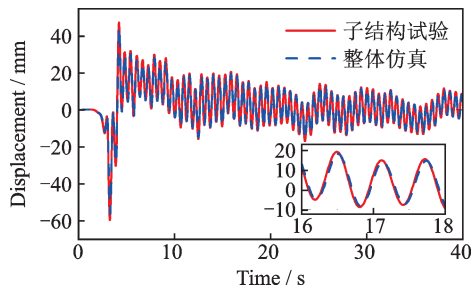


图 7 1500 自由度 $\Delta t=4 \text{ ms}$ 试验子结构位移时程

Fig. 7 Displacement time history of physical substructure with 1500 DOFs when $\Delta t=4 \text{ ms}$

说明该试验架构能满足实时子结构试验的精度需要。表 4 中工况 1~8 对应的试验与整体仿真时程对比如图 8 所示。图 8 为 $\Delta t=4, 20 \text{ ms}$ 时,实时子结构试验中测试所得物理子结构位移时程与整体仿真结果对比。8 组工况条件下基于 GPU 的子结构试验与整体仿真结果一致,说明使用 GPU 求解数值模型的子结构试验与 CPU 具有同样精度,实现了与 CPU 试验性能相同但数值计算体量更大的实时子结构试验。

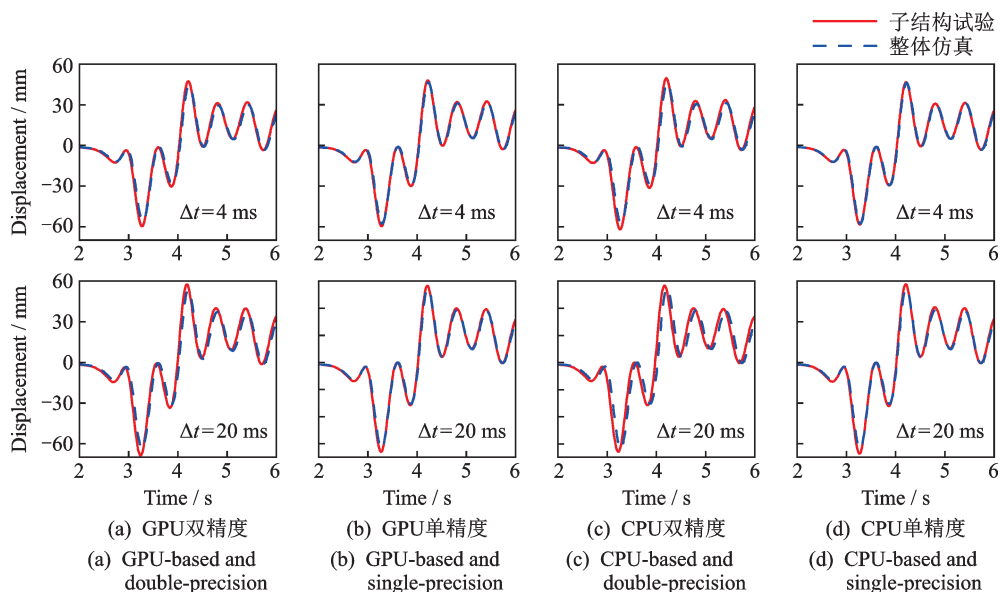


图 8 实时子结构试验与整体仿真实物理子结构位移对比

Fig. 8 Displacement of physical substructure from RTHT testing and integral simulation

除计算自由度需求外,数值子结构求解精度也直接影响实时子结构试验性能。由于各数值积分算法只能求得近似解,其求解精度随积分步长减小而增大。因此,如何通过减小积分步长提高数值积分求解精度和稳定性,是数值子结构求解要面临的另一个问题。为了讨论积分步长对子结构试验性能的影响,采用 3888 自由度的数值子结构模型,使用表 1 中 CPU 时最小只能进行步长为 20 ms 的实时计算,使用 GPU 时最小实时计算步长为 5 ms。对 $\Delta t=5, 20 \text{ ms}$ 时分别进行了实时子结构试验。所得试验子

结构顶部位移时程与整体仿真结果如图 9 所示。由图 9 可知, $\Delta t=5 \text{ ms}$ 子结构试验结果与整体仿真结果峰值误差为 4.92%, $\Delta t=20 \text{ ms}$ 试验结果与仿真结果峰值误差为 10.28%。由此可见,采用 GPU 求解可以允许在实时子结构试验中使用更小的时间步长,从而进一步提高试验精度。

3.3 试验结果分析

由图 8(a) 即表 4 工况 1~4 条件下 $\Delta t=4 \text{ ms}$, 双精度条件下, GPU 求解数值模型最大自由度数为

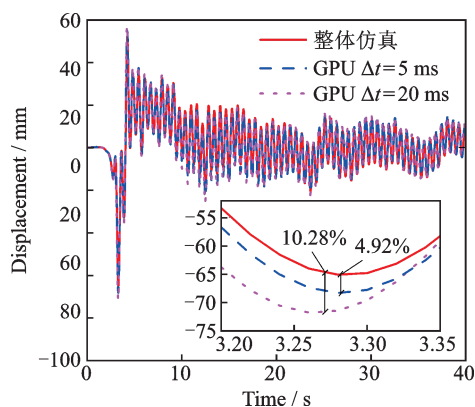


图9 不同积分步长下实时子结构试验位移对比

Fig.9 Real-time testing displacement of physical substructure with different time steps

1500,使用CPU求解数值模型最大自由度数为1080,GPU求解的优势并不明显。当使用单精度数据求解时,GPU求解最大自由度数为3168,CPU求解最大自由度数为1500,可见使用单精度数据GPU求解的优势相比双精度更明显。目前因Windows系统不稳定性及LABVIEW软件调用MATLAB软件的方法通讯需要3ms的问题有待解决,因此在较小步长时GPU的优势受到限制,此方法调用GPU求解时存在改进优化空间。

由图8(b)即表4工况5~8即 $\Delta t=20$ ms条件下,使用双精度数据GPU求解最大自由度可达18876,此时CPU求解自由度仅为2904。使用单精度数据GPU求解最大自由度可达27000,使用CPU求解自由度仅为3888。可见在 $\Delta t=20$ ms时,使用GPU求解优势明显,无论是单精度数据还是双精度数据都远超CPU求解时的数值模型自由度。使用GPU求解可以实现相同时间步长条件下CPU无法求解的大规模数值模型的实时子结构试验。

实时子结构试验中数值模型求解需满足积分算法收敛性要求,使用本文配置的服务器在 $\Delta t=20$ ms条件下,基于MATLAB软件GPU工具箱最多可进行27000自由度的实时子结构试验。相同数值模型使用CPU时间步长需936ms,使用GPU已实现相比CPU求解55倍加速效果,此时CPU已经远不能满足实时子结构试验的实时性需要。本试验求解是基于MATLAB软件的GPU工具箱进行求解,受限于MATLAB软件,GPU求解的性能还有进一步提升的余地,在GPU并行运算、资源配置、及硬件数据通讯等方面还可进行优化,有可能进一步提升求解规模与效率。

表4中数值子结构模型为3888自由度时,使用GPU求解相比CPU求解时间步长从20ms减小至5ms。且由图9相同数值模型,进行不同时间步长

试验对比可以得出,使用更小的时间步长可以提高试验精度。因此在实时子结构试验中,相同数值子结构使用GPU求解代替传统CPU求解可以减小时间步长,提高试验精度。

4 结 论

本文基于GPU数值求解建立了实时子结构试验系统架构,通过数值仿真与振动台子结构试验,验证了该试验系统的可行性和实际试验性能,得到如下结论:

(1) 在相同的时间步长条件下,基于MATLAB软件使用GPU相比CPU可求解更高自由度数值子结构。在 $\Delta t=20$ ms条件下,使用GPU可实现27000自由度数值模型的实时子结构试验。

(2) 在相同自由度的数值模型条件下,基于MATLAB软件使用GPU相比CPU求解实现更小时间步长的实时子结构试验。在3888自由度条件下,使用CPU求解可实现最小 $\Delta t=20$ ms的实时子结构试验,而使用GPU求解可实现的试验最小 $\Delta t=5$ ms。

(3) 由于Windows是非实时系统,以及使用LABVIEW调用MATLAB软件存在信号传输延迟问题,因此在试验中最小取 $\Delta t=4$ ms,这限制了GPU性能的发挥。有必要研究基于硬件实时系统的GPU求解方法,彻底发挥GPU求解的性能,进一步拓展实时子结构试验的使用空间。

参考文献:

- [1] Nakashima M, Kato H, Takaoka E. Development of real-time pseudo dynamic testing [J]. Earthquake Engineering & Structural Dynamics, 1992, 21(1): 79-92.
- [2] 吴斌,王倩颖.实时子结构实验的研究进展[J].实验力学,2007(6):547-555.
Wu Bin, Wang Qianying, Development of real-time substructure testing [J]. Journal of Experimental Mechanics, 2007(6): 547-555.
- [3] 孟凡涛,赵建锋,于广明.实时子结构混合试验中的数值积分方法对比分析[J].地震工程与工程振动,2011,31(5):60-67.
Meng Fantao, Zhao Jianfeng, Yu Guangming. Study on numerical integration methods in real-time hybrid testing experiment [J]. Earthquake Engineering and Engineer Vibration, 2011, 31(5): 60-67.
- [4] Zhu F, Wang J, Jin F, et al. Simulation of large-scale numerical substructure in real-time dynamic hybrid testing [J]. Earthquake Engineering and Engineering Vibra-

- tion, 2014, 13(4): 599-609.
- [5] Wang Q Y, Wu B, Ou J P. Stability analysis of real-time substructure testing considering actuator delay and compensation [J]. *Engineering Mechanics*, 2007, 24(2): 9-14.
- [6] 洪越, 唐贞云, 何涛, 等. 大尺寸非线性实时动力子结构试验实现 [J]. *振动工程学报*, 2017, 30(6): 913-920.
Hong Yue, Tang Zhenyun, He Tao, et al. The implementation of nonlinear real-time dynamics substructuring for large scale specimen[J]. *Journal of Vibration Engineering*, 2017, 30(6): 913-920.
- [7] Chang S Y, Sung Y C. An enhanced explicit pseudo dynamic algorithm with unconditional stability [C]. 100th Anniversary Earthquake Conference, 2006.
- [8] Nakashima M, Ishida M, Ando K. Integration techniques for substructure pseudo dynamic test: pseudo dynamic test using substructuring techniques [J]. *Journal of Structural & Construction Engineering*, 1990, 417: 107-117.
- [9] Wu B, Xu G, Wang Q, et al. Operator-splitting method for real-time substructure testing [J]. *Earthquake Engineering & Structural Dynamics*, 2010, 35(3): 293-314.
- [10] Nakashima M, Masaoka N. Real-time on-line test for MDOF systems [J]. *Earthquake Engineering & Structural Dynamics*, 1999, 28(4): 393-420.
- [11] Cheng C, Ricles J M. Stability analysis of SDOF real-time hybrid testing systems with explicit integration algorithms and actuator delay [J]. *Earthquake Engineering & Structural Dynamics*, 2008, 37(4): 597-613.
- [12] Chae Y, Kazemibidokhti K, Ricles J M. Adaptive time series compensator for delay compensation of servo-hydraulic actuator systems for real-time hybrid simulation [J]. *Earthquake Engineering & Structural Dynamics*, 2013, 42(11): 1697-1715.
- [13] Saouma V, Kang D, Haussmann G. A computational finite-element program for hybrid simulation [J]. *Earthquake Engineering & Structural Dynamics*, 2012, 41(3): 375-389.
- [14] Durand M, Marin P, Faure F, et al. DEM-based simulation of concrete structures on GPU [J]. *European Journal of Environmental & Civil Engineering*, 2012, 16(9): 1102-1114.
- [15] 解琳琳, 韩博, 许镇, 等. 基于 OpenSees 的大型结构分析 GPU 高性能计算方法 [J]. *土木建筑工程信息技术*, 2014, 6(5): 22-25.
Xie Linlin, Han Bo, Xu Zhen, et al. GPU powered high-performance computing method for the analysis of large-scale structures based on OpenSees [J]. *Journal of Information Technology in Civil Engineering and Architecture*, 2014, 6(5): 22-25.
- [16] 武唯一, 田石柱, 李双江. 基于 ABAQUS 与 OpenFresco 的结构抗震混合试验平台的研究 [J]. *江苏建筑*, 2016(4): 21-25.
Wu Weiyi, Tian Shizhu, Li Shuangjiang. Study of Structure seismic hybrid simulation based on ABAQUS and OpenFresco [J]. *Jiangsu Construction*, 2016(4): 21-25.
- [17] 许国山, 郝伟, 陈永盛, 等. 基于 OpenFresco-LabVIEW-dSPACE 的混合试验系统研究 [J]. *工程力学*, 2013, 30(3): 417-423.
Xu Guoshan, Hao Wei, Chen Yongsheng, et al. Experimental validation on hybrid testing system based on OpenFresco-LabVIEW-dSPACE [J]. *Engineering Mechanics*, 2013, 30(3): 417-423.
- [18] Tang Z, Dietz M, Hong Y, et al. Performance extension of shaking table-based real-time dynamic hybrid testing through full state control via simulation [J]. *Structural Control and Health Monitoring*, 2020, 27(10): e2611.

Performance verification of GPU-based framework for real-time hybrid testing

DONG Xiao-hui¹, TANG Zhen-yun^{1,2}, LI Zhen-bao¹, DU Xiu-li¹

(1. The Key Laboratory of Urban Security and Disaster Engineering of Ministry of Education, Beijing University of Technology, Beijing 100124, China; 2. Hebei Key Laboratory of Earthquake Disaster Prevention and Risk Assessment, Sanhe 065201, China)

Abstract: A RTHT framework based on LABVIEW-MATLAB-GPU is established, which uses graphics processor (GPU) instead of the traditional central processing unit (CPU) as the hardware for numerical solution. Taking a soil-structure interaction system as an example, the performance of the testing framework is verified through numerical simulation and RTHT testing. The testing and simulation results demonstrate that the proposed RTHT framework allows numerical substructure with 27000 DOFs to be solved in real time, hence that the application range of RTHT is expanded significantly by the calculation capacity of GPU.

Key words: real-time hybrid testing; graphics processing unit; numerical integration algorithm; shaking table; finite element analysis

作者简介: 董晓辉(1997—),男,硕士研究生。电话:15503181183;E-mail:dongxiaohui@emails.bjut.edu.cn。

通讯作者: 唐贞云(1983—),男,副研究员,博士生导师。电话:13810980957;E-mail:tyz@bjut.edu.cn。