



基于WebGL和有限元的 码头三维设计与计算仿真

张丽媛, 苏永生, 芦志强

(中交水运规划设计院有限公司, 北京 100007)

摘要: 在复杂的码头工程中, 三维设计与有限元计算仿真不仅能够解决Autocad图纸表现力不足的问题, 还能提高设计效率, 为结构分析和优化设计提供有力参考。针对码头平面设计难以与有限元计算衔接、三维有限元模型难以用于交流的问题, 将有限元与WebGL技术结合起来, 开发出一套3D设计仿真一体化程序。该程序依托Autocad、Adina平台, 从二维图纸出发, 能够半自动化地生成有限元计算文件, 该文件可以直接提交运算, 得到应力、温度等各项指标。计算结果经过进一步处理能够在网络平台展示计算结果、实现界面交互。该套技术方案实现了3D设计、计算、展示的统一, 成功地应用于工程实例, 取得了良好的效果。

关键词: 码头; WEBGL; 有限元; 工程可视化

中图分类号: U 656

文献标志码: A

文章编号: 1002-4972(2013)12-0057-07

3D design and simulation of wharf based on WebGL and FEM

ZHANG Li-yuan, SU Yong-sheng, LU Zhi-qiang

(CCCC Water Transportation Consultants Co., Ltd., Beijing 100007, China)

Abstract: In complicated wharf engineering, 3D-based design and FEM not only enhance performance of 2D drawing, and improve efficiency, but also offer strong reference to structure analysis and optimization. In order to better connect autoCAD drawing to FEM calculation and enable FEM file to communicate on the internet, FEM is combined with WebGL and a set of programmes for 3D design and analysis is developed. The programmes are run on Autocad and Adina platform, and start from 2D drawing, build FEM files semi-automatically, and the stress and temperature field are derived. The postfile is further processed so that we can display contour plots and execute interactive operation on internet. The whole set of technical solution integrates design, calculation and display. It is tested and verified on a project with satisfactory results.

Key words: wharf; WebGL; FEM; VCAD

长久以来, 以CAD为基础的2D模型在码头工程设计中占据举足轻重的地位, 3D模型一般用于工程展示或是满足复杂问题的有限元计算需要。但是近些年, 基于3D的设计理念逐渐浮出水面, 以Autodesk、PTC、MicroStation为代表的工程软件公司先后开发出基于3D设计的软件或设计模块。这种设计理念的优势是明显的: 能够更

直观、形象地表达模型; 方便各专业人员之间的协调合作与交流, 修改也更有效率; 自动生成图纸, 减少因人工出错的概率。但上述3D设计、分析面临两个主要问题, 一是造模软件与有限元软件不能够有效衔接, 格式转换有时会造成模型失真或数据错误; 二是目前的3D模型依赖于本地计算机或局域网服务器, 不能实现实时、有效的交

收稿日期: 2013-03-21

作者简介: 张丽媛(1988—), 女, 硕士, 助理工程师, 从事港口工程、内河工程等工作。

流。对于问题一,笔者早先的工作已经在水工重力坝结构中实现AutoCAD图纸到有限元计算的衔接^[1],本文通过对EAM软件加以改进,将其应用于码头工程。对于问题二,基于广域网络平台的3D设计显然更具备普遍性和发展前景。事实上新一代的HTML5标准推出后,WebGL便迅速发展。后者为前者提供了3D渲染接口,其卓越的渲染能力和丰富的渲染功能大大突破了在网页中开发3D模型的限制。因此讨论3D模型设计、分析、展示一体化,并应用于广域网平台,符合码头工程的设计发展趋势,具有一定的应用意义。

本文在Autocad和Adina平台下,使用lisp语言、EXCEL VBA、Fortran语言开发了一套自动化程序,从CAD图纸出发,根据用户需求半自动地生成有限元计算文件,利用ADINA求解器进行结构分析。有限元分析结果经过WebGL程序自动处理生成用于网络交流的3D模型,该模型不仅能够动态展示分析对象,同时能显示结构分析结果,如应力场云图、温度场云图等。整套程序将二维图纸与有限元计算有效衔接,并将有限元计算结果与WebGL在网络中的展示功能结合起来,实现了上述设计、计算仿真与网络展示一体化功能。

1 概述

1.1 有限元方法与应用

有限元方法起源于20世纪50年代,是一种数值计算方法,广泛应用于各种微分方程描述的场问题求解。它的基本思想为:通过离散求解域,使用分片连续函数近似整体未知场函数(插值函数或位移函数),基于基本方程,建立起单元节点的平衡方程,使用矩阵代数求解联立方程组,求解连续场问题。随着计算机硬件水平的不断提高,有限元方法逐渐成为工程领域设计、分析的主要工具。它能够处理具有普通或复杂几何、荷载和边界条件的大型问题,甚至实现动态施工仿真。目前,工程领域常用的大型商用有限元软件有ADINA, ANSYS, ABAQUS, MARC等。

商用有限元软件以求解器为核心,进行工程分析的一般流程为:前处理、计算、后处理。其中,前处理包括建立模型、划分网格、赋予材料

属性、添加边界及荷载条件、设置工况等操作。以往基于2D的设计,由于面对从2D图纸到3D模型的转换、协调边界网格,赋予材料属性、施加荷载及边界条件等大量手工操作,前处理在整个有限元分析流程中最为费时。商用软件提供的简化操作的手段主要有命令流和python脚本两种,前者以ANSYS, ADINA为代表,后者以Marc, Abaqus为代表。

码头工程中常涉及岩土相关计算,因此本文依托ADINA平台进行开发。由于上述有限元软件均拥有丰富的接口,模型可以跨平台使用,因此在ADINA平台下开发出的模型在其他软件中也可以方便地使用。

1.2 HTML5下的WebGL

HTML5是取代了HTML4.01和XHTML1.0标准的HTML标准版本,其开发技术包括HTML, CSS和Javascript在内的一套技术组合。HTML5优势明显^[2],尤其在Web网页的表现性和应用程序功能上,因而逐渐受到Chrome, Opera, Firefox等主流浏览器的充分支持。

WebGL是一个基于OpenGL ES2.0的3D渲染API^[3],可以为HTML5的Canvas提供GPU加速,能够在网页上创建3D模型,使得HTML不必再依赖第三方工具进行复杂模型的绘制和动态图片展现。WebGL在HTML5的Canvas中可实现的功能多种多样,例如交互操作,为模型添加颜色、材质、光照效果等,甚至实现场景漫游、动画等等。

在HTML5中开发WebGL模块涉及的程序语言主要有3种:Json, ESSL (OpenGL ES Shading language), Javascript。其中,Json是一种轻量级、跨语言的文本型数据交换格式,主要用于传递模型数据,整个过程通过Javascript中内嵌的Json对象实现^[4]。ESSL用于为顶点着色器和片元着色器编写实现具体算法^[5],语法结构与C/C++类似。Javascript是一种基于对象的脚本语言,可以直接嵌入HTML文档中或者动态装载,使得网页拥有丰富多彩的动画和用户交互,大大拓展了网页的功能^[2]。

1.3 WebGL中的颜色与光照

在WebGL中,利用颜色和光照可以有效区

分不同地质、结构、材料,更直观地展示模型、增强3D视觉效果,最终的视觉效果是两者综合作用的结果。对于颜色和光照效果,WebGL中有不同的模式可供选择。常用的着色方法有3种:Flat,

Goraud, Phong, 分别对应基于多边形、顶点、像素计算光照效果的方法。三者之间的对比见表1^[6-7]。尽管Goraud插值方法具有自身的局限性,但它仍是现今硬件最常用的方法^[8]。

表1 计算结果与色彩对应关系

着色方法	插值方法	优缺点
Flat	不插值,每个三角形采用均一光光照值	计算量小;光滑性差,无光泽感
Goraud	对定点颜色插值确定像素颜色值	计算量小,光滑性较好;高光连续性较差
Phong	对顶点法向量插值计算多边形各像素的法向量,确定每个像素颜色值	光照和雾化效果好;计算量大

常用的光照效果(镜面反射)模型为Lambertian模型和Phong模型。前者认为物体最终呈现的颜色只受物体和光散射性质的影响,后者则综合考虑了背景光、物体(和光)散射性质、反射性质的影响,因此,Goraud着色配合Phong镜面反射模型是较为可行的颜色光照表现方法。

本文在ESSL中使用如下代码实现两者的结合:

$v_{\text{FinalColor}} = I_a + I_d + I_s;$

$I_a = u_{\text{LightAmbient}};$

$I_d = u_{\text{LightDiffuse}} * u_{\text{MaterialDiffuse}} * \text{lambertTerm};$

$I_s = u_{\text{LightSpecular}} * u_{\text{MaterialSpecular}} * \text{SpecularCoefficient}$

$\text{lambertTerm} = \text{dot}(\text{Normal_vertex}, -\text{Direction_light});$

其中: N 为顶点法向量; L 为光照方向向量; I_a 为环境光分量; I_d 为散射分量; I_s 为反射分量。显然Phong模型考虑更为全面。

1.4 界面交互

界面交互包括平移、旋转、缩放等基本操作,方便用户查看分析3D模型,更进一步可实现场景漫游。界面交互的函数通过javascript来实现。

本文表2中所示函数、变量控制鼠标和键盘,在页面中实现旋转、平移、缩放的交互操作。

2 开发流程

基于有限元和WebGL技术的特点开发的3D设计、计算、展示一体化程序,流程如图1所示。该流程包含5个关键步骤:

1)从Autocad二维图纸提取模型信息。本步骤通过Autocad的lisp脚本辅助完成。

表2 WebGL中界面交互函数

项目	类型	控制内容
ONMOUSEUP、 ONMOUSEDOWN、 ONMOUSEMOVE	函数	鼠标
ONKEYUP ONKEYDOWN MOVE、ROTATE、ZOOM	函数	键盘
POSITION AZIMUTH ELEVATION	变量	观察者视线位置、视线方向角、视线仰角
DX、DY	变量	鼠标的移动增量
PACEX、PACEY、 PACEZ、SCALE	变量	显示幅度

2)整合提取的模型信息,编制ADINA可执行的命令流文件。该流文件可直接在ADINA GUI或DOS下执行得到3D有限元模型。本步骤通过EXCEL VBA, FORTRAN, 批处理文件辅助完成。

3)有限元计算。该步骤在有限元软件求解器中完成。

4)从有限元计算结果中提取节点、单元拓扑、材料组单元等信息整合成Json文件格式。本步骤通过笔者编制的FORTRAN程序辅助自动完成。

5)将Json文件加载到HTML5模板中,生成可以用于在网络环境中展示与交流的文件。该模板中WebGL接口负责对模型渲染和界面交互,相关代码已在1.3和1.4节中叙述。由于该段代码无需更改,并且Json文件名称固定,程序可自动判别,本步骤自动完成。

综上所述,该方法以Autocad图纸为起点,最终得到三维有限元模型及计算结果以及可以用于在网络环境中展示与交流的HTML5文件。全部操作由笔者编制的程序包半自动化地完成,大大提

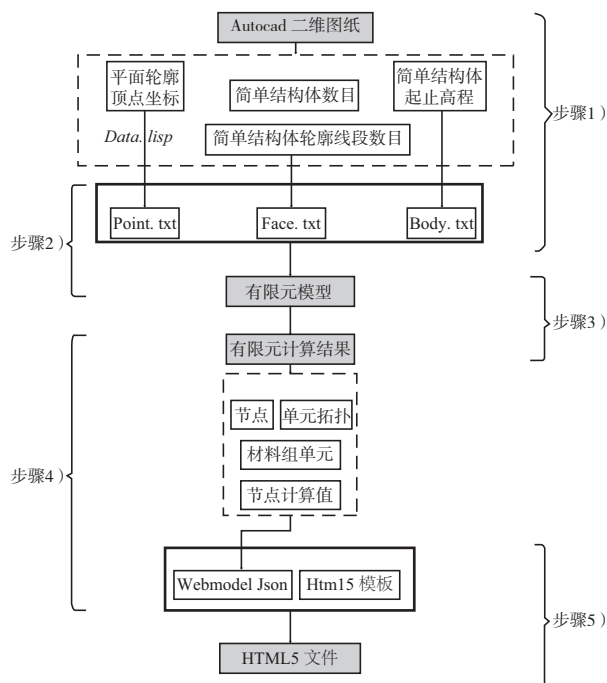


图1 2D到3D的HTML5文件的开发框架

高了分析效率。该程序包在笔者早先的工作^[1]基础上进行改进,主要模块及其实现方法在第3节中详细叙述。

3 主要模块实现方法

3.1 从CAD到有限元模型

第2节所述步骤1)和2)实现了从Autocad图纸到有限元模型的过渡。三维建模按照由点到线、由线到面、由面到体的顺序进行。首先,用户需要对Autocad文件中的分析对象进行结构识别,即确定简单结构体数目。分析对象由多个简单结构体组成,简单结构体的特征是其剖面图为闭合曲线。以图2所示重力式码头为例,以整体为分析对象,共包含12个简单结构体:胸墙、7个沉箱(7个)、碎石基床、地基、2种回填土。

运行data.lsp提取每个简单结构体的轮廓线的坐标,由于data.lsp在平面图形上提取数据,因此所有点在第3方向上的坐标为0。Data.lsp提取坐标函数如下:

```
(setq k (getstring "\n输入点号 (如: P1) ! ")
p (getpoint "\n选取坐标点! ")
x (strcat "X=" (rtos (nth 0 p)))
y (strcat "Y=" (rtos (nth 1 p))))
```

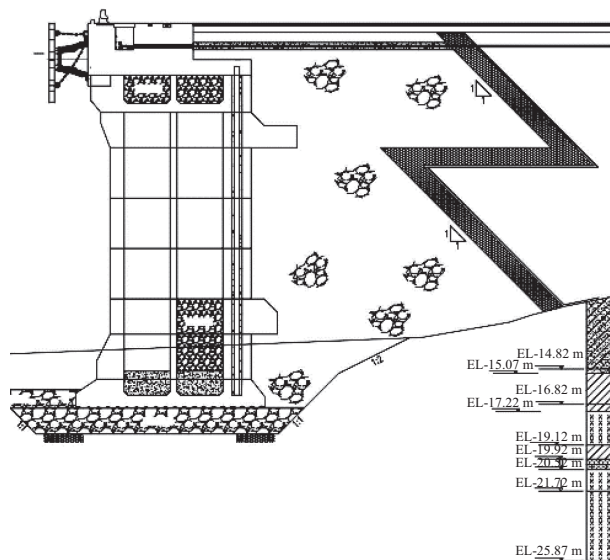


图2 重力式码头结构示意图

全部提取完毕后所有数据将汇总至coordinates.dat文件,将数据拷贝到EAM程序包中的EXCEL表格1,并修改各数据第3方向上的坐标值。修改完毕后,EXCLE环境下VBA程序会生成模型所需的point.txt文件。EXCEL下的运行界面如图3所示。

结构体编号	点序号	X	Y	Z	备注
1	1	-80400	-32300	0	0 抛填, 10.45-37.99
	2	-69900	-32300	0	
	3	-69900	-20300	0	
	4	-70900	-20300	0	
	5	-72600	-18627	0	
	6	-72600	-2100	0	
	7	-69900	-2100	0	
2	8	-69900	-100	0	0 上游孔, 18.98-37.99
	9	-69400	-100	0	
	10	-74100	-4200	0	
	11	-73600	-3700	0	
	12	-73600	-2100	0	
	13	-74100	-1600	0	
	14	-78900	-1600	0	
	15	-79400	-2100	0	
	16	-79400	-3700	0	
	17	-78900	-4200	0	

图3 Excel表格1运行界面

点击“下一步”,Excel程序自动跳转到表格2。将统计好的简单结构体数目、每个简单结构体轮廓线段数填入并确定,即生成face.txt;填入每个简单结构体在第3方向上的起止位置,即可生成body.txt。至此,应用ADINA平台建模所需的全部数据文件已经生成,按照提示退出Excel。运行程序包根目录下的execute.bat文件即可将生成的point.txt,face.txt和body.txt在dos环境中整合成ADINA命令流文件(.in文件)。在ADINA GUI环境中加载

该文件即可看到模型效果, 针对图2所示结构, 建模效果如图4所示。

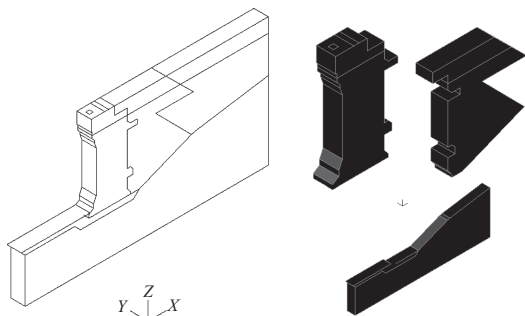


图4 有限元模型欧整体及局部示意图

计算前, 还需要进行划分网格, 对不同的材料、结构体划分组别, 加载边界条件、荷载等操作。上述操作需要在ADINA平台下手动完成。由于用于计算的有限元模型应使用3D网格对实体划分, 用于网络的模型应使用2D网格对实体表面划分, 为操作方便, 建议首先进行3D网格划分, 再使用convert-face-to-element方法将3D网格的表面拓扑转化为2D单元, 计算时不激活所有的2D单元。

3.2 网络环境下有限元结果的云图显示

步骤4) 是有限元计算结果的展示, 如绘制温度场、应力场云图。这需要建立节点计算值与色彩之间的映射关系。色彩通过色彩坐标表现, 每种颜色在不同的色彩模型中对应不同的坐标值, 同种颜色在不同模型中的坐标值可以互相转换。建立映射关系的方法为: 在色彩域 $[C_1, C_n]$ 中线性插值 $n-1$ 个点, 每个点对应一个颜色坐标值, 同时将值域 $[V_{\min}, V_{\max}]$ 划分为 n 段, 每一段 $[V_{m-1}, V_m]$ 均采用 C_{m-1} 颜色表示^[9], 其中划分段数 n , C_1, C_n 可以由用户确定。彩虹色阶 (Rainbow platte) 是较为常用的色域, 以RGB模型为例划分5段色阶为例, 其对应方法如表3所示^[10]。

表3 计算结果与色彩对应关系

计算值/最大值	R	G	B
1.0~0.8	255	0→255	0
0.8~0.6	255→0	255	0
0.6~0.4	0	255	0→255
0.4~0.2	0	255→0	255
0.2~0.0	0→255	0	255

由于计算完毕的有限元结果文件不仅包含了描述模型的节点、单元拓扑、单元组, 还包含各节点的温度值、应力值, 各部分信息使用关键字进行分割, 因此使用程序找出计算值对应的关键字, 按照上述方法将节点计算值映射到RGB值中非常方便。本套程序能够自动完成并映射, 并将映射好的颜色值一并其他信息按照单元组别整理成一系列Json文件, 同时在Json文件中添加属性uPervertexcolor=true, 即设置WebGL的渲染模式为顶点染色模式。每个顶点对应4个浮点数以表示其颜色值: R,G,B,Alpha, Alpha代表透明度。在Json格式中, 各部分的对应关系为: 节点-vertices, 单元拓扑-indices, 转换后的温度、应力值-colors。

对于未经计算的有限元模型, 模型文件中不含有计算结果, 但仍保留节点、单元拓扑、单元组等信息。本套程序在进行整理时, 会自动设定uPertexclor=false, 即采用环境光(diffuse)对不同材料组进行上色, 以区分结构不同区块。

3.3 Json文件加载

步骤5) 是对Json文件的装载到HTML5模板, 并生成最终展示文件的过程, 该模板部分样式集和表单借鉴了文献[6]中提供的代码。

步骤4) 生成的Json文件含有模型网格顶点坐标和索引表, WebGL通过这两类信息描述模型, 同时通过Json文件中的colors或diffuse属性给模型上色。加载过程所用函数为:

1) loadObjectByParts/ loadObject: 读取Json文件;

2) JSON.parse: Javascript内置函数, 该函数将每个Json文件的描述对象创建为类, Json文件中的关键字定义为类属性;

3) Scene.addObject: 链接缓存IBO、VBO、NBO等。

最终生成的HTML5文件函数调用流程如图5所示。

其中, WebGLAPP代表WebGL在HTML5中的Javascript接口, Configure函数用于初始化页面参数, 如设定canvas背景色等; Load函数用于加载Json模型文件; Draw函数用于渲染模型。

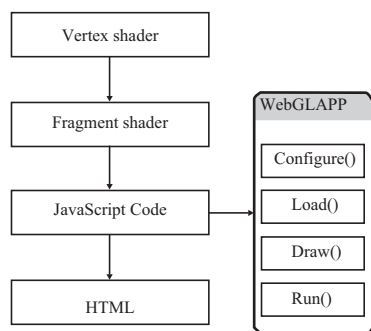


图5 嵌入WebGL的HTML 5开发流程

4 应用实例

沙特达曼港集装箱泊位工程规模位于阿拉伯半岛东部, 沙特东海岸波斯湾达曼港西港池, 拟建2个集装箱泊位。水工采用重力式空心方块结构, 岸壁高度约为21 m, 方块质量150 t以内, 方块成列布置, 采用钢轨整体串联, 每4列方块顶部采用现浇胸墙连接。方块空心处根据计算需要适当回填块石。为降低墙厚回填块石量, 墙厚棱体呈闪电状。上部集装箱岸桥规矩30.48 m, 前轨位于重力墙身上, 后轨采用轨道梁结合灌注桩支撑体系, 注桩直径1 m, 间距4.2 m。系船设施分别为150 t系船柱和H1600锥形护舷。单位长度结构的有限元模型展示效果如图6所示, 运行环境为firefox 18.0.2。



图6 沙特达曼重力式方块码头结构示意

为增加对比效果, 图6使用灰度对不同结构及材料加以区分, 区分内容包括地基、抛填块石、空心方块混凝土、胸墙混凝土、回填料、混凝土等。图中能体现明显的光照效果。同时用户能够运用鼠标、键盘实现缩放、旋转、移动等交互操作。

沙特地区气候炎热, 并且现场使用混凝土方块体积较大, 浇筑后实测温度较高。施工中混凝土温度控制是否达标关系到结构安全, 因此温控问题一度成为关注的重点。为此, 笔者对典型方块进行了温度应力耦合有限元计算, 典型方块尺

寸及连接方式如图7所示。

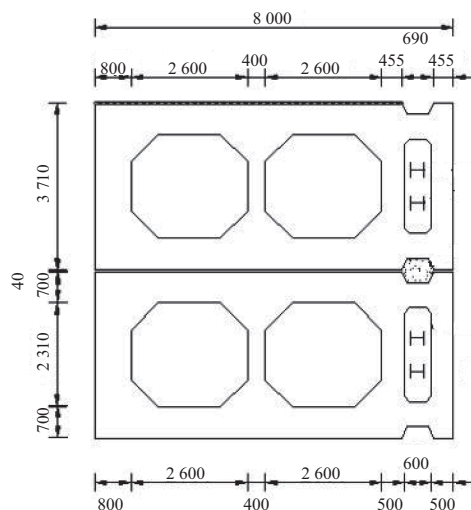


图7 典型方块尺寸及连接方式 (单位: mm)

拆模时刻, 有限元计算出的温度、应力结果如图8和9所示。从图中可以看出, 环境温度为23℃, 混凝土内部最高温度约为65℃。能够明显看出, 地面温度相对较低, 混凝土方块内部温度

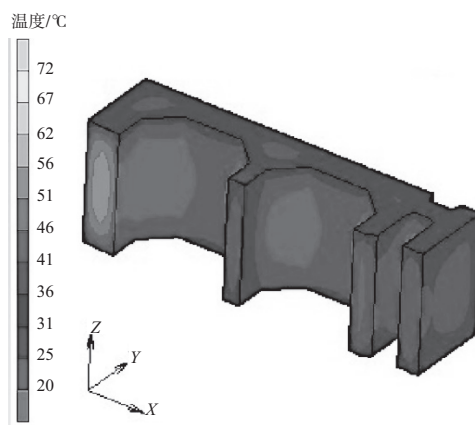


图8 混凝土方块有限元计算温度值

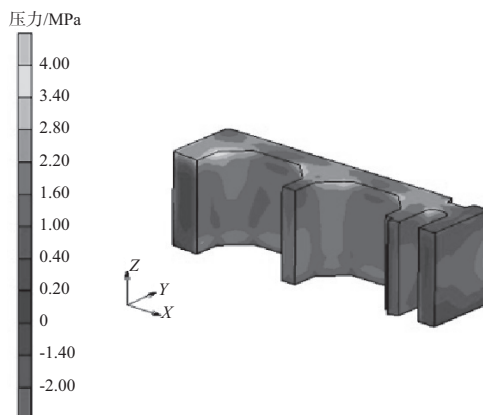


图9 混凝土方块有限元计算应力值

较高。方块厚壁内侧与环境温差较大,具有较大的安全风险,通过应力计算,该局部在未考虑配筋情况下,拉应力约为3 MPa,尚属安全。

对有限元计算结果进行处理后,稳定温度场云图在网页上的展示效果如图10所示,图中颜色由浅至深表示温度由高到低。此时,混凝土温度与环境温度的差异已明显减小。

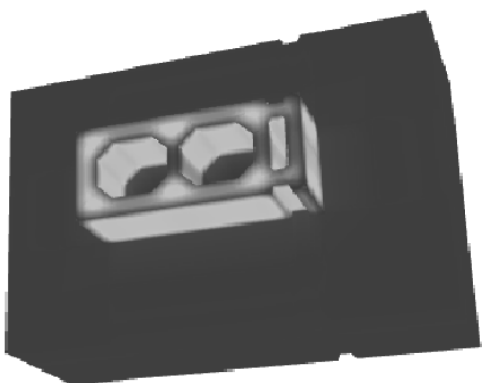


图10 温度模拟云图

5 结语

为了顺应码头3D设计的大趋势,解决Autocad图纸表现力不足、难以有限元软件衔接等问题,同时提高设计效率、方便优化设计与交流,结合码头工程的特点和实际需要,将有限元技术与WebGL技术结合起来,开发出一套3D设计仿真一体化程序。该套程序从Autocad图纸出发,能够半自动化地生成ADINA平台下的有限元计算文件,该文件可以直接提交运算。计算结果经过进一步处理能够在网络平台展示云图、实现界面交互。该套技术方案实现了3D设计、计算、展示的统一,并成功地应用于工程实例,取得了良好的效果。本文提出的技术解决方案能够拓展到水运工程的其他领域。

事实上,从结构整体安全、沉降计算到温度应力仿真,有限元在码头工程中的应用已经越

来越广泛,但对于WebGL的研究和应用还尚属初步,本文提出的方法虽能够满足基本需要,但还有很大的深入空间。随着研究深入,笔者认为可以增加对如下模块的开发:与数据库结合进行施工过程动态模拟展示、对有限元计算结果进行动态演示、对静态图像增加切片效果等等。

参考文献:

- [1] 张丽媛,胡昱,李庆斌,等.基于平面布尔运算的重力坝3D有限元建模[J].水力发电学报,2012,31(5):209-215.
- [2] 李刚.疯狂html5、CSS3、Javascript讲义[M].北京:电子工业出版社,2012.
- [3] khronos. WebGL specification[EB/OL]. (2013-08-07) [2013-08-12] <http://www.khronos.org/registry/webgl/specs/latest>.
- [4] Crockford D. The application/json media type for javascript object notation (json)[J]. The Internet Society, 2006(7): 1-8.
- [5] Kessenich J D, Baldwin R. Rost. The opengl shading language[M/OL]. Indiana, USA: Addison-Wesley Professional, 2004. <http://www.opengl.org/documentation/glsl>.
- [6] Cantor D. WebGL Beginner's Guide[M]. Birmingham, United Kingdom: Packt Publishing, Limited, 2012.
- [7] 孙家广,胡事民.计算机图形学基础教程[M].北京:清华大学出版社有限公司,2005.
- [8] Dunn, F, I. Parberry, 3D math primer for graphics and game development[M] Birmingham, United Kingdom: AK Peters Limited, 2011.
- [9] 张师帅,罗军.三维温度场计算机可视化系统的研制与开发[J].热能动力工程,1999,14(6):441-442.
- [10] Brewer C A. Color use guidelines for data representation[C]// Proceedings of the Section on Statistical Graphics. Alexandria: American Statistical Association, 1999:55-60.

(本文编辑 武亚庆)