PRE-PROCESSING OF LIGHTING HEAT ANALYSIS AT VISTEON-AUTOPAL

Tomas Havlik*, Roman Malucha
Visteon-Autopal, Czech Republic

KEYWORDS - heat analysis, lighting, ANSA, script, I-Deas TMG/ESC

ABSTRACT - Software I-Deas TMG/ESC (hereinafter TMG) is used by Visteon-Autopal for the heat analysis of lighting. Ansa is used for pre-processing and TMG is used for following solving and post-processing. Script for acceleration work created in the TMG (hereinafter TMGscript) allows the fast setting of the complete process but it requires modifications of the script for each analysis, therefore Ansa scripting language (hereinafter Ansascript) was chosen in preference.

The main function of Ansascript is to transfer the FE model from Ansa to TMG and to pre-process the analysis. Ansascript recognizes element property names which contain keywords for part names, materials, boundary conditions, etc., and accordingly, creates sets in Nastran deck and TMGscripts which use these sets. The transferring process is to run the TMGscripts created by Ansascript which work with the FE model saved by Ansascript as a TMG universal file.

The reason we use ANSA scripting language and not TMG scripting language is that Ansascript is easier and based on language “C” which is found in most manuals. The process of transferring using Ansascript, reduces the pre-processing time, simplifies model modification and eliminates mistakes made when setting boundary conditions and materials, manually. This procedure can be applied to any other software.

TECHNICAL PAPER -

1. INTRODUCTION

Visteon-Autopal is one of the leader automotive lighting companies. During development of last 10-15 years we could have seen quite significant changes – plastic materials mostly removed metal and glass material from lighting product by increasing of their heat resistance. And escalation of competition between lighting suppliers decreased time and cost for development as much as possible. Simulations happened important and necessary tool in supplier contest.

Figure 1 – History and present

2. HEAT ANALYSES

Heat simulation plays significant role in lighting product development. Components are under quite big heat load from operated bulbs. The trend in design points to using the smallest dimensions of lamp chambers with the highest wattage source because of more original appearance. More heat resistant but more expensive materials do not meet low cost requests. A few heat simulations have to be performed to find a compromise - optimal design and material in short time.

Heat simulation is necessary to perform during concept phase, when changes are easy to do. Bad concept could lead to the lamp that need not to meet customer requests (tests) and could increase overall cost. Lamp design is complicated and consist many design features and components that are not important from point of view heat simulation. So CAD data has to be simplified. ANSA provides us the performance that we need to build a high quality FE-Model in short time.

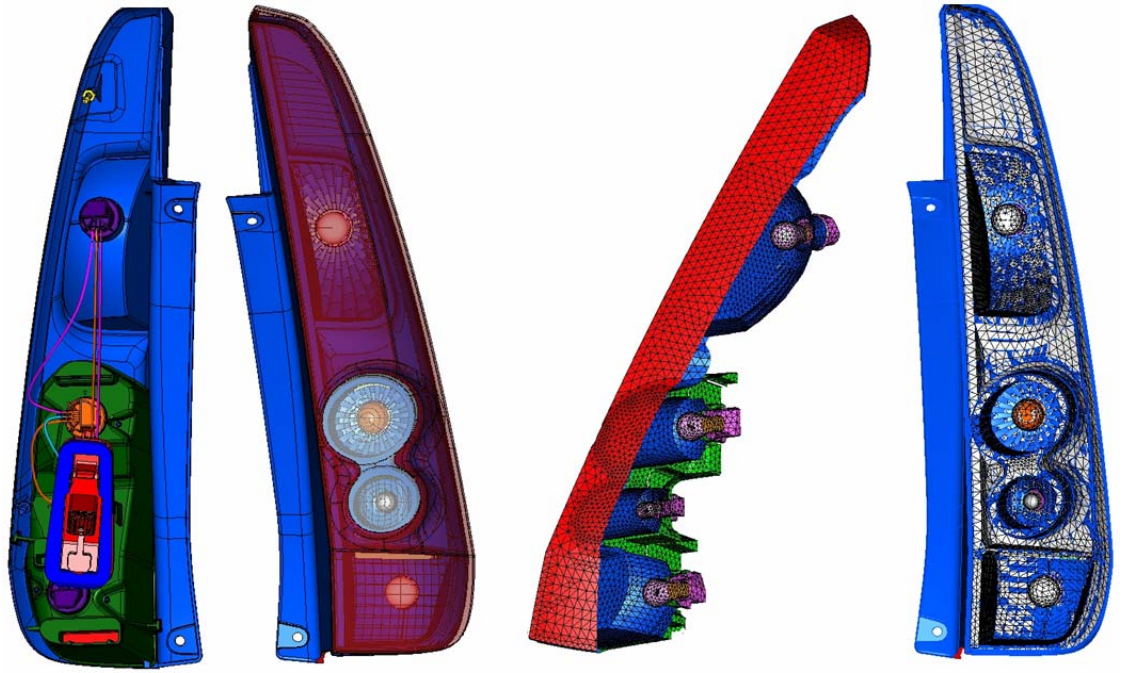

Figure 2 – CAD data to FE-Model transfer

TMG heat transfer solver can solve all three principles of heat transfer in automotive lighting – radiation, convection and conduction.

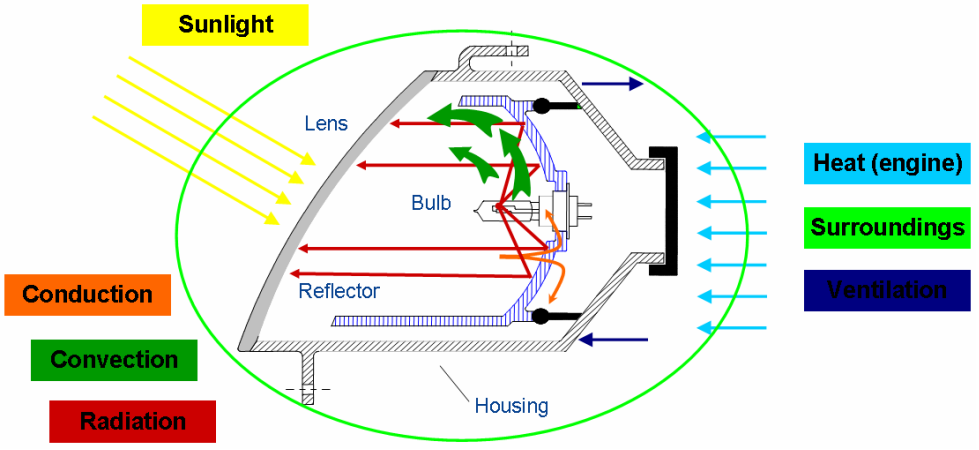

Figure 3 – Three principles of heat transfer in automotive lighting

3. PRE-PROCESSING

Ansa supports output of FE-Model to I-Deas universal format. Boundary conditions can be set in DECK>NASTRAN but TMG boundary conditions and materials are not supported. After FE-Model is done in Ansa, setting of boundary conditions and materials is made in TMG. This hand work can cause a lot of mistakes. For example you can spend time by setting materials in TMG and in the end of your work you find mistake so you repair the model in Ansa and set it again in TMG or you want do some modification. The same situation, you have to do settings all again. For that reason some automation is required.

First automation was made using the TMGscript. It used numbers of group (set in Ansa) and it was not user friendly language. When we wanted set TMG it took lots of time to change syntax of the TMGscript, but it allowed us further modifications of the model and eliminated mistakes. It was great start and useful for next automation process.

As a second automation we needed universal script which we didn't have to change for each calculation. As the best was Ansascript. There is shown how we use Ansascript in figure 4.

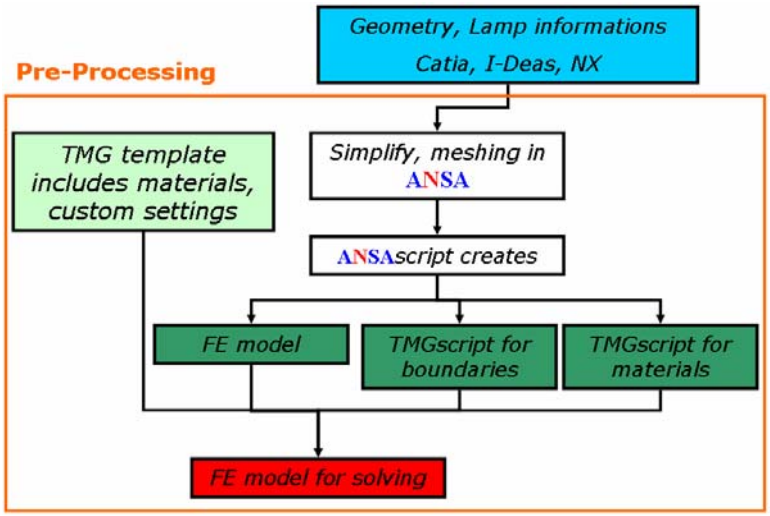

Figure 4 – Pre-Processing Heat Analysis

We have predefined template file with material library and custom settings. We start from that template and load FE model (File 1) which includes groups. After loading we execute two TMGscripts (Files 2, 3) for setting boundary conditions and materials. These three files are made by Ansascript. Ansascript works with property names (Figure 5).

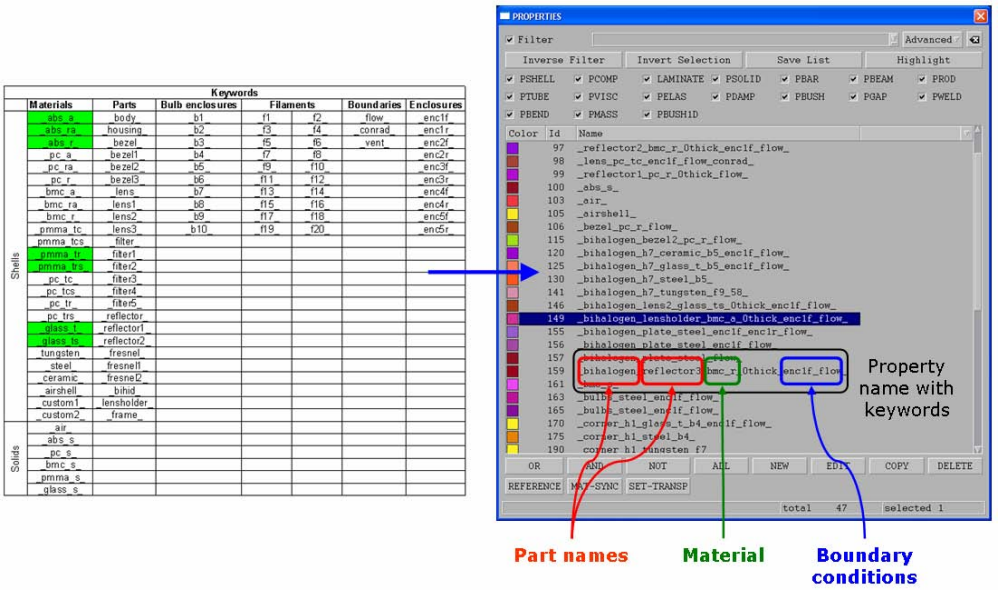

Figure 5 – Defined keywords

We defined keywords what we use for element property (PIDs) names. Ansascript seeks for a keyword in PID name and creates set from all PIDs which includes this keyword in PID name. Simultaneously Ansascript writes to TMGscript needed syntax. After Ansascript seeks all keywords, TMGscripts are saved and FE-Model is saved to TMG universal file. For better orientation we separated material and boundary conditions settings to two TMGscripts. Function which creates set from all PIDs include keyword is in figure 6.

```
//-----------------------------------------------------------------------------
//Function for making groups according to parameters
//-----------------------------------------------------------------------------

def MakeSetFromPid (string name,int id,string TypeOfElement)
{
    search_type[0] = "P" % TypeOfElement;
    pelements = CollectEntities(NASTRAN,0,search_type,0);                  //Collect all PSHELLS

    matrix wattages = {"3.0", "3.8", "5.5", "6.6", "8.26","8.8" ,
                       "14.6" ,"17.8" ,"18.55" ,"18.6", "19.2", "19.4",
                       "21.24" ,"23.6", "23.9" ,"26.25", "26.5", "36",
                       "43", "58", "68"};

    k=0;

    watt = 1;

        foreach pelement in pelements
        {
            GetEntityCardValues(NASTRAN,pelement,"Name",prop_name);        //Get the name of each PSHELL
            ind = MatchString(prop_name,name);                             //Check for the name that it was given as input
            if(ind)
            {

                foreach wattage in wattages
                    {
                    if (MatchString(prop_name, wattage))
                    watt = wattage;
                    }

                to_set[k]=pelement;                                        //Store the PSHELLs that contain the name
                k++;
            }
        }
    if(k==0)
    {
        Print("None");
        return 0;
    }
    search_type[0] = TypeOfElement;
    if (TypeOfElement == "BEAM")
        {
        search_type[0] = "CBEAM";
        }
    elements = CollectEntities(NASTRAN,to_set,search_type,1);              //Collect the shells that use the PSHELLs that were found
    set = CreateEntityVa(NASTRAN,"SET","Name","SID",id);                   //Create a set
    AddToSet(set,elements);                                                //Add to the created set the shells

    Print("Done");
    return watt;
}

//-----------------------------------------------------------------------------
```

Figure 6 – Main function of Ansascript

4. CONCLUSION

This process can be applied to any other software so in future we will use it for NX 5. Advantage of the Ansascript is that it is based on language “C” which is found in most manuals. Using of the Ansascript reduces time of Pre-Processing, reduces mistakes made by human and lets easy modifications and makes Pre-Processing more funny.