

АВТОМАТИЗАЦИЯ ИЗВЛЕЧЕНИЯ ДАННЫХ ИНЖЕНЕРНОГО АНАЛИЗА ПРИ МОДЕЛИРОВАНИИ ЯЧЕЙСТЫХ СТРУКТУР

Ярмошук Ю.М., Юхо Е.Н.

Научный руководитель – Полозков Ю.В., к.т.н., доцент

В процессе оптимизации топологической формы деталей путем встраивания ячеистых структур на изменение физико-технических свойств существенное влияние оказывают количество, геометрическая форма и размещение ячеек [1, 2]. В связи с этим для анализа напряженно-деформированного состояния деталей посредством проведения итерационного моделирования и выполнения инженерных расчетов деталей с ячеистыми структурами было разработано программное обеспечение с использованием APISolidWorks [2, 3]. По окончании выполнения расчетов с помощью этого программного обеспечения в качестве выходных данных выступают значения напряжений, которые возвращаются посредством метода `APIGetMinMaxStress (intNComponent, intNElementNumber, intNStepNum, objectDispPlane, intNUnits, outintErrorCode)` и значения деформаций, возвращаемых посредством метода `APIGetMinMaxDisplacement (intNComponent, intNStepNumber, objectDispPlane, intNUnits, outintErrorCode)`. Указанные методы возвращают значения напряжений и деформаций, рассчитываемые для каждого из узлов конечно-элементной сетки, которая строится на основе твердотельной модели детали. Причем, могут быть получены значения, соответствующие величине нормалей к плоскостям пространственной системы координат, векторам, инцидентным плоскостям пространственной системы координат, а также значения эквивалентных напряжений по теории Мизеса. В зависимости от плотности сетки эти значения могут исчисляться десятками и сотнями тысяч.

Полученные данные изначально хранятся в виде сплошного массива значений номеров узлов и действующих на эти узлы 11 типов нагрузок. Для удобства последующей обработки и наглядного представления извлеченные данные необходимо структурировать и отобразить в графическом виде, например, в виде графиков или диаграмм.

Для создания удобных списков объектов был разработан метод `FillingTotalStressList`, который разделяет исходный массив напряжений на лист объектов `TotalStressModel`, а также аналогичный метод `FillingTotalStrainList` для разделения исходного массива деформаций на лист

объектов TotalStrainModel. Для структурированного вывода этих листов в виде таблиц используется компонента DataGridView. Источником данных для компоненты указывается лист классов. Результат полученных преобразований представлен на рисунке 1.

Strain	Stress									
NodeId	SX	SY	SZ	XY	XZ	YZ	ESTRN	SEDENS	ENERGY	
1	-8,76637762E-10	1,01199431E-08	-2,684792E-09	-3,934833E-09	4,38868941E-10	-1,42086618E-08	1,16856782E-08	2,896263E-07	0	
2	-8,718338E-10	5,73048275E-09	-1,16228638E-09	-2,09687512E-09	-8,15281731E-10	-1,29652014E-08	8,830782E-09	1,55022164E-07	0	
3	-5,491503E-10	2,7719298E-09	-5,482782E-10	-1,50193225E-09	-8,232831E-10	-1,08875469E-08	6,737322E-09	7,05406649E-08	0	
4	-2,12303022E-10	1,39303791E-09	-3,62860353E-10	-9,526582E-10	-3,85594445E-10	-9,63424E-09	5,70565462E-09	4,53781261E-08	0	
5	-5,41017474E-11	4,62741845E-10	-2,40132636E-10	-8,419461E-11	-2,9642E-10	-8,763164E-09	5,07997777E-09	3,44356472E-08	0	
6	4,21075154E-11	-2,47070225E-10	-1,52205165E-10	7,254216E-10	-2,37116576E-10	-8,202913E-09	4,75945239E-09	3,01559133E-08	0	
7	1,882513E-10	-9,594103E-10	-1,07113131E-10	1,53492985E-09	-2,503829E-10	-7,75891E-09	4,620242E-09	2,96881044E-08	0	
8	3,4985792E-10	-1,76488535E-09	-5,887896E-11	2,47549736E-09	-3,016149E-10	-7,48345652E-09	4,73477524E-09	3,37983366E-08	0	
9	5,20632371E-10	-2,93869618E-09	1,13890425E-11	3,53403284E-09	-4,01784273E-10	-7,34586436E-09	5,182194E-09	4,6893863E-08	0	
10	6,286339E-10	-4,62585525E-09	2,57884158E-10	4,92346652E-09	-5,04989E-10	-7,966375E-09	6,386375E-09	8,12599836E-08	0	
11	6,37634834E-10	-8,1221625E-09	2,03609352E-09	5,60915669E-09	-1,60909661E-10	-7,865663E-09	8,457944E-09	1,60594837E-07	0	
12	1,807276E-09	-9,075609E-09	4,14782431E-09	2,7487066E-09	-1,10939091E-10	-5,16990051E-09	8,821916E-09	1,35659974E-07	0	
13	1,93324179E-09	-6,172559E-09	2,842664E-09	-6,38054748E-11	3,76096515E-10	-2,61487E-09	5,93070926E-09	5,16923144E-08	0	
14	1,643988E-09	-4,53942262E-09	1,90329974E-09	-5,915425E-10	1,35661413E-10	-2,241733E-09	4,419679E-09	2,81470172E-08	0	

Рисунок 1 – табличное представление листа класса извлечения значений деформации

Для визуализации данных применяется фреймворк LiveChart. К проекту его можно подключить через встроенную систему NuGet. Отличительными особенностями графиков этого фреймворка являются: встроенное отображение значения точки, при наведении на кривую; сглаженные графики; возможность обновления графиков в текущем времени. За отображение графиков отвечает абстрактный класс, основной функцией которого является возможность преобразования последовательности значений в кривую. Основные методы класса, отвечающего за стартовые настройки графиков и создание кривых, представлены ниже:

```
virtualpublicvoid InitChart()
{
if (IsInitialize()) //проверка количества графиков
{
GetChart.AxisX.Clear(); //отчистка стандартной оси X
GetChart.AxisX.Add(new Axis //создание новой оси X
```

```

        {
IsEnabled = true, //видимость
ShowLabels = true, //отображение текста
Separator = newSeparator//создание вертикальной сетки
        {
StrokeThickness = 1, //толщина линии
//цвет линии
Stroke = new System.Windows.Media.SolidColorBrush(System.Windows.Media.Color.FromRgb(64,
79, 86))
        },
Labels = newList<string>()
        });
GetChart.AxisY.Clear(); //отчистка стандартной оси Y
GetChart.AxisY.Add(newAxis//создание новой оси Y
        {
IsMerged = true, //отображение текста внутри сетки
Separator = newSeparator//создание горизонтальной сетки
        {
StrokeThickness = 1, //толщина линии
//цвет линии
Stroke = new System.Windows.Media.SolidColorBrush(System.Windows.Media.Color.FromRgb(64,
79, 86))
        }
});
GetChart.Zoom = ZoomingOptions.X; //функция увеличения
GetChart.LegendLocation = LegendLocation.Top; //положение подписей кривых
}.

public LineSeriesAddSerie(string name)
{
LineSeriesser = new LineSeries() //создание новой кривой
{
Values = new ChartValues<float>(), //тип значений

```

```

PointGeometry = null, //отображение точек на графике
Fill = System.Windows.Media.Brushes.Transparent, //окрашивание области под кривой
StrokeThickness = 1.5, //толщина линии
Title = name//подпись кривой
    };
return ser;
}.

```

Пример преобразования точек в кривую:

```

public void FillingChartPart(int mass, int start, int id, int ser, string serName)
    {
ClearValues(ser); //отчищаем значения кривых
GetChart.AxisX[0].Labels.Clear(); //отчищаем подписи осей X

int end = start + mass; //поиск последней точки вывода относительно положения ползунка
//выход за пределы массива точек, вывод до конца
if (end > TotalStrainModels[id].Count)
    {
end = TotalStrainModels[id].Count;
    }

//цикл вывода части графика
for (int startP = start; startP < end; startP++)
{
//выбор по имени кривой
switch (serName)
    {
case "XY":
        {
//преобразование значения массива в точку кривой
GetChart.Series[ser].Values.Add(TotalStrainModels[id][startP].XY);
//добавление подписи точки

```

```

GetChart.AxisX[0].Labels.Add(TotalStrainModels[id][startP].NodeId.ToString());
break;
}
case "XZ": ...

```

Результатом работы рассмотренного класса является динамическое отображение графика или части графика (рисунок 2).

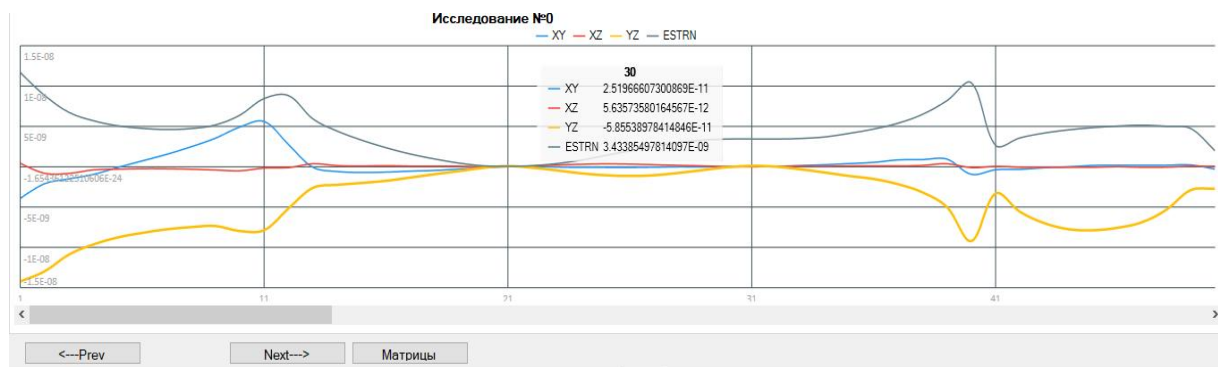


Рисунок 2 – графики некоторых табличных значений: напряжения по осям и общее напряжение

Для управления отображением графиков создана панель включающая: чекбоксы отвечающие за подключение кривых, которые строятся на основе данных из таблиц, текстбоксы для задания масштаба – количество точек кривой, переход к отрезку графика, в который входит указанная точка и интервал и ряд других функциональных элементов управления.

Представленные программные средства структурированного представления и визуализации данных способствуют выработке дальнейших решений в процессе автоматизации анализа данных о напряженно-деформированном состоянии деталей с целью выявления наиболее рациональных областей размещения и геометрической конфигурации ячеистых структур при проектировании легковесных деталей.

Литература

1. Полозков, Ю.В. Проблемы проектирования и формообразования легковесных деталей в аддитивном производстве / Ю.В. Полозков // Математические методы в технике и технологиях : сб. тр. междунар. науч. конф., Минск, 10 – 12 октября 2017 г. / СПб.: Изд-во Политехн. ун-та ; под общ.ред. А. А. Большакова. – Минск, 2017. – Т. 10 – С. 61 – 65.

2. Полозков, Ю.В. Автоматизация оценки влияния конфигурации ячеистых структур на физико-технические свойства детали / Ю.В. Полозков, Е.Н. Юхо, С.А. Рагуля, А.Ю. Калита // Информационные технологии и системы: проблемы, методы, решения (ИТС – 2018) : сб. материалов Республиканской научно-технической конференции, Минск, 23 – 24 ноября 2018 г. / Четыре четверти; редкол.: С. В. Харитончик [и др.]. – Минск : 2018. – С. 124 – 129.

3. SOLIDWORKS API Help [Electronic resource] / DassaultSystèmes – Mode of access : <http://help.solidworks.com/2019/english/api/sldworksapiproguide/Welcome.htm/> – Date of access : 22.08.2018.