

# Расчёт обобщённых угловых коэффициентов излучения для зонального метода с использованием средств графического конвейера

И.А. Лебедев,  
м.н.с., [LebedevMSZ@gmail.com](mailto:LebedevMSZ@gmail.com),  
ИПУ РАН, г. Москва

В статье описывается новый подход к расчету угловых коэффициентов излучения, необходимых при моделировании тепловых процессов зональными методами. Принципиальной особенностью нового алгоритма является использование графического конвейера для большей части вычислений. Суть метода заключается в замене источника излучения «наблюдателем» и оценке угловых размеров объектов системы, попадающих в поле его зрения. Тем самым определяется, какая часть излучения, распространяемая из источника, попадает на эти объекты, то есть определяются угловые коэффициенты излучения. Верификация предложенного способа проводится путем сравнения показаний штатных управляющих термопреобразователей реальной купольной вертикальной вакуумной электрической печи сопротивления с аналогичными показаниями её математической модели распределения тепловых потоков с использованием предложенного алгоритма расчета угловых коэффициентов излучения.

The article describes a new approach to the calculation of angular radiation coefficients required for modeling of thermal processes zonal methods. The principal feature of the new algorithm is to use a graphics pipeline (a standard means of rendering images) for most of the calculations. The method consists in the backward compatibility of the terms "source" and "receiver" - if the radiation falls on the first element from the second element of the system, then the radiation from the first element to the second element falls. Replacing the radiation source "observer" and estimate the angular size of the objects that fall into his field of vision, you can determine which part of the radiation propagating from the source falls on the objects, that is, to determine angular radiation coefficients. This method allows to calculate the slopes of the system of arbitrary complexity with any required accuracy. Tools DirectX help automate and conceal from the programmer most of calculations greatly simplify the writing and to reduce runtime. Using a shader computing avoids wasting time copying data from the video memory to the memory of a personal computer. A simple modification of this method allows to apply it to a mirror reflection.

## Введение

Данная статья является продолжением статьи [1]. В ходе дальнейших исследований была проведена оптимизация алгоритма расчета обобщенных угловых коэффициентов и проведена его верификация.

Угловой коэффициент излучения – это величина, характеризующая поток излучения, который попадает с одной площадки изучаемой системы на другую. Если  $i$ -тая излучающая поверхность (источник излучения) системы излучает на  $j$ -тую излучающую поверхность системы (приёмник излучения), то  $\varphi_{i,j}$  определяет, какая часть от общего излучения  $i$ -той поверхности попадает на  $j$ -тую.

Если принять излучение полностью диффузионным (независимым от направления), то угловой коэффициент отражает, какая часть излучения источника, распространяемого по полусфере, попадает на проекцию приёмника на эту сферу. Используя термины трехмерной компьютерной графики, получается, что если поставить наблюдателя в точку источника излучения, и направить взгляд по направлению нормали в данной точке и отобразить полученное изображение в сферической системе координат (зенитные и азимутные углы откладываются по осям  $x$  и  $y$ ,  $z$  выполняет роль глубины изображения – координаты  $z$ ), то отношение полученной площади приёмника к полной площади изображения, есть угловой коэффициент излучения.

## Постановка задачи

Суть предложенного способа заключается в представлении исследуемого объекта (вакуумного печного агрегата) в виде трехмерной модели (в виде последовательности вершин в декартовой системе координат и информации о связях между вершинами). В трехмерной модели объекта поочередно каждая вершина, принимаемая как источник излучения, представляется в виде «камеры» и прорисовывается видимое ей изображения средствами графического ускорителя в сферической системе координат и обработке этого полученного изображения, с учетом того факта, что вместо цвета пиксель должен содержать информацию о номере излучающей поверхности элемента системы. Полученное в ходе обработки изображение неинформативно для вывода на экран, поскольку не имеет цветов и сохраняется только в текстуру в видеопамяти. Обработка изображения имеет целью рассчитать степень «видимости» остальных элементов системы из точки «камеры».

Из этого следует, что для расчета углового коэффициента излучения необходимо:

- 1) Поставить наблюдателя (камеру) в точку, принимаемую как локальный источник излучения, таких точек на одной поверхности может быть сколько угодно, для увеличения точности расчета, при этом интенсивность каждой отдельной точки обратно пропорциональна их количеству.
- 2) Направлением камеры выбрать нормаль к излучающей поверхности в данной точке.
- 3) Произвести рендеринг изображения, переведя перед этим все координаты в сферическую систему координат. В цвете поступающих на отрисовку примитивов закодирован номер изотермической зоны и номер излучающей поверхности этой зоны.
- 4) Определить на полученном изображении отношение площадей видимых поверхностей к площади изображения (точнее круговой центральной его части, так как координаты точек будут иметь разброс по первым двум координатам от  $-\pi/2$  до  $\pi/2$ ).

## Реализация на основе графического конвейера

Графический процессор это отдельное устройство в составе компьютера, которое выполняет обработку графической информации для вывода на экран (рендеринг). Благодаря специализированной конвейерной архитектуре он справляется с этой задачей намного лучше центрального процессора. Видеоадаптеры (видеокарты) имеют в своем составе до нескольких тысяч ядер и собственную оперативную память, их архитектура оптимизирована для массивных параллельных вычислений, что и определяет их большую эффективность в рендеринге. Графический конвейер – это аппаратно-программный комплекс визуализации трёхмерной графики, последовательность стадий (функций), выполняющихся в фиксированном порядке и параллельно. Каждая стадия принимает информацию из предыдущей стадии и отправляет в следующую. Стандартный графический конвейер обрабатывает вершины, геометрические примитивы, а также пиксели конвейерным способом. В отличие от центрального процессора, графический процессор имеет большое количество ядер, что позволяет ему обрабатывать высокий объем данных параллельно, не ставя их в очередь. В современном графическом конвейере поддерживаемом в DirectX 11 пять программируемых стадий: вершинный шейдер(VertexShader), шейдер поверхности(HullShader), шейдер домена (DomainShader), геометрический шейдер (GeometryShader) и пиксельный шейдер (PixelShader).

Вершинный шейдер вызывается для каждой вершины. В данной реализации служит для перевода локальных координат примитивов (треугольников) в глобальные, а затем из глобальных в координаты камеры. В итоге координата z отвечает за глубину изображения, то есть, направлена вглубь экрана. Особенностью кода является то, что вместо цвета, для дальнейшей обработки передается номер элемента системы, к которому принадлежит вершина (в данном случае это два числа - номер изотермической зоны и номер излучающей поверхности этой зоны).

Так как графический конвейер способен работать только с вершинами в декартовой системе координат, то он будет некорректно проводить растеризацию, то есть заполнения пространства внутри треугольника. Прямые, соединяющие вершины в декартовой системе координат, не являются таковыми при отображении в сферической, вырождаясь в дуги, что вызывает погрешности, вплоть до полного искажения результатов. Решением данной проблемы в рамках графического конвейера может быть увеличение числа треугольников, то есть уменьшение их размеров. Но задавать заранее большее количество вершин накладно с точки зрения ресурсов и отнимает у алгоритма гибкость (необходимо загружать разные модели для разной точности расчетов). Более правильным является использование двух следующих стадий конвейера, появившихся в 11 версии DirectX: шейдер поверхности и шейдер домена. Их совокупная работа позволяет организовать так называемую тесселяцию – разделение поступающих на вход треугольников на более мелкие треугольники «на лету», т.е. генерируемые вершины не присутствуют в исходной модели, а появляются на этой стадии. Это позволяет резко повысить детализацию трехмерной модели, в том числе и в сферической системе координат, и следственно точность расчета угловых коэффициентов. Чем больше уровень тесселяции тем точнее грани первичного треугольника соответствуют их реальному виду в сферической системе координат. Данный способ позволяет организовать изменение уровня тесселяции в зависимости от расстояния до объекта и его углового размера, так как нет смысла тратить ресурсы графического процессора на деление далеких от камеры треугольников, они и так отобразятся близко к истине, в то время как близкие объекты будут искажены, и требуют более высокого уровня тесселяции. Так же тесселяция позволяет на «на лету» округлять поверхности, что удобно при прорисовке сфер, цилиндров и других сферических объектов, так как они при разбиении на излучающие поверхности заменяются правильными многоугольниками и многогранниками.

В качестве примера необходимости применения тесселяции рассмотрим простейший случай – куб, каждая из граней которого состоит из двух треугольников и является излучающей поверхностью. Если поставить камеру в центр одной из граней и смотреть внутрь куба то в сферических координатах получится картинка, представленная на рисунке 1. Каждая грань для удобства раскрашена своим цветом. Рядом для наглядности каркасное изображение.

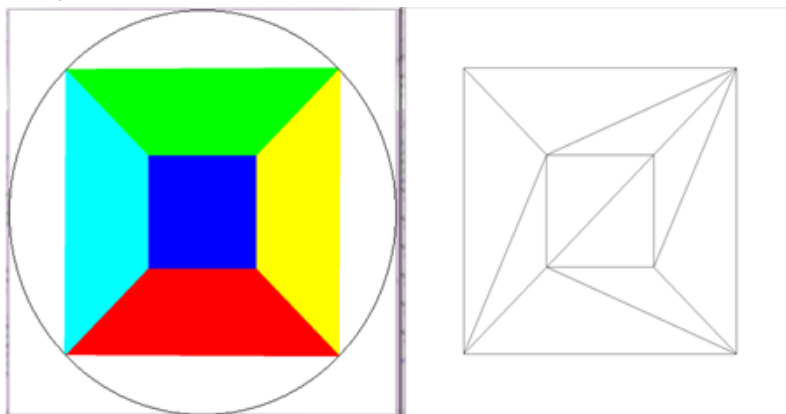


рис. 1 Отображения куба в сферических координатах без тесселяции

Как видно из представленного рисунка, получившееся изображение сильно искажено слишком малым количеством вершин. Соединяя их по прямой при рендеринге графический процессор не учитывает что это сферическая система координат, в результате мы получим сильно заниженные угловые коэффициенты и часть излучения будет просто «улетать в пустоту» (в случае куба очевидно что излучение изнутри никуда не может исчезнуть и должна быть заполнена вся центральная круговая зона). При тесселяции каждая из сторон исходного треугольника и его внутренняя область разбивается на несколько отрезков, которые становятся сторонами новых более мелких треугольников. Назовем количество делений уровнем тесселяции. На рисунках 1-3 представлены изображения при уровнях тесселяции 0, 2 и 10 соответственно (рендеринг производился в текстуру 3000x3000 точек).

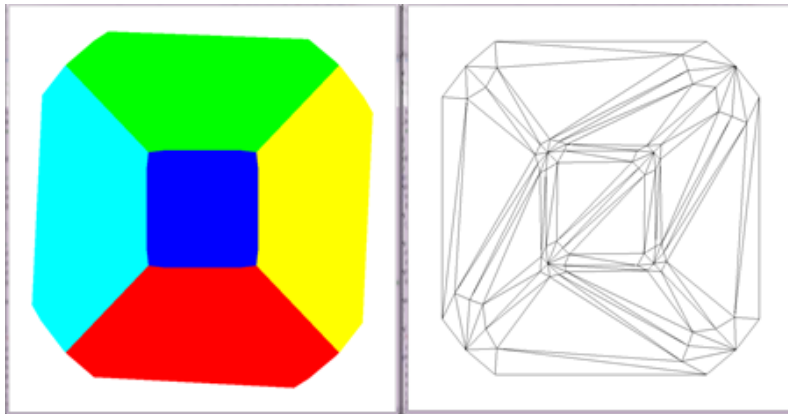


рис. 2 Отображения куба в сферических координатах с уровнем тесселяции 2

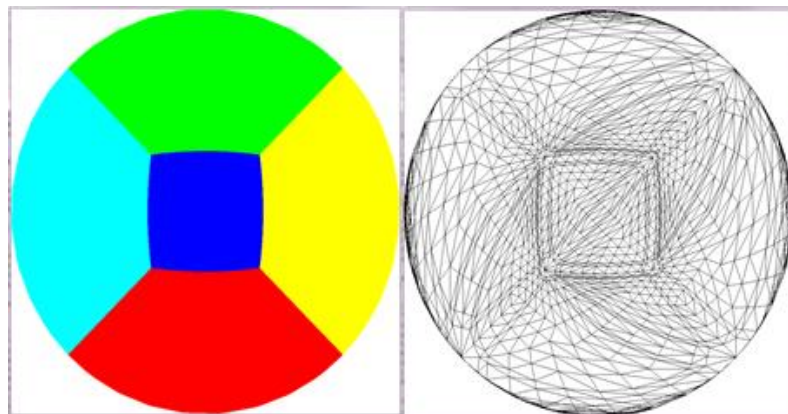


рис. 3 Отображения куба в сферических координатах с уровнем тесселяции 10

Полученные вершины поступают на следующую стадию – геометрический шейдер. В отличие от вершинного он обрабатывает весь примитив целиком, т.е. на его вход приходит сразу три вершины, принадлежащие одному треугольнику. Так же этот шейдер способен генерировать новые произвольные (по заданному пользователем алгоритму) вершины и удалять старые. На этом этапе необходимо сначала проверить вершины на видимость камерой, и если необходимо пересчитать некоторые из них. Видимой для камеры вершиной в данном случае считается вершина с неотрицательной координатой  $z$  (вся полусфера). В случае если в треугольнике нет неотрицательных вершин, он не отправляется на дальнейшую отрисовку (вершины удаляются).

На этапе растеризации графический процессор закрашивает все пиксели, принадлежащие треугольнику, но при стандартном использовании графического конвейера применяется матрица проекций, трансформирующая координаты примитивов в проекционные. Координата  $z$  (глубина изображения) приводится к диапазону  $(0.0, 1.0)$ , а координаты  $x$  и  $y$  приводятся к виду, в котором видимая часть изображения имеет координаты  $(-0.5, 0.5)$ .

В сферической системе координат по оси  $z$  откладываем  $\theta \cos(\varphi)$ , по оси  $y$  -  $\theta \sin(\varphi)$ ,  $r$  – выступает в роли глубины изображения, где  $\theta$  - зенитный угол,  $\varphi$  - азимутный угол,  $r$  - кратчайшее расстояние до начала координат (камеры). Для компенсации отсутствия проекционной матрицы необходимо разделить  $r$  на некоторую, заранее большую максимального расстояния в исследуемой модели величину (таким образом, приведя к нужному диапазону); в 4 компоненту координаты вершины в сферической системе координат вписать  $\pi/2$  (так как диапазон сферических координат  $(-\pi/2, \pi/2)$ ).

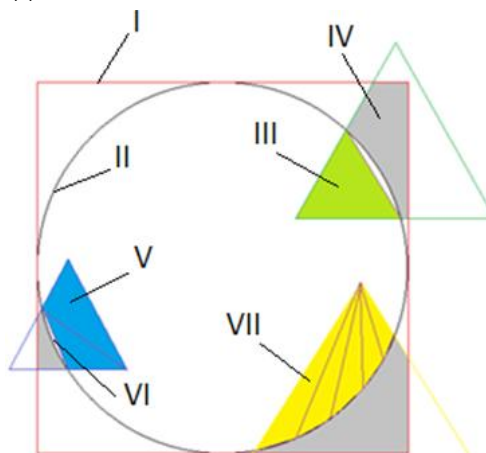


рис. 4 Пример отрисовки

На рисунке 4 приводится пример прорисовки треугольников. Красный квадрат определяет зону выводимую на экран (кадр), черный круг показывает разброс сферических координат. В случае зеленого треугольника имеется одна неотрицательная вершина, при перерасчете в сферические координаты будет закрашена сверх необходимого серая зона, что приведет к неправильному расчёту площади. Необходимо заменить каждую из двух точек за границей обзора на точки пересечения с окружностью (в декартовой системе это точки пересечения с плоскостью XY). В случае наличие двух неотрицательных точек (синий треугольник) заменяется только одна из точек, но количество треугольников удваивается.

Белый участок между отрисованным треугольником и окружностью обзора появляется из-за погрешности приведения к сферической системе, и исправляется тесселяцией (желтый треугольник). После выполнения геометрического шейдера треугольник отобразится внутри зоны ограниченной кругом, площадью  $\pi/4$  от общей площади квадратного изображения. Ниже приведен код перевода из декартовых в сферические координаты. Если треугольник полностью лежит в неотрицательной области z, то он отрисовывается без изменений. В полученной системе точки, лежащие на одной прямой, если смотреть из начала координат (камеры) имеют одни и те же координаты x и y, но разную составляющую z, что позволяет применить стандартный буфер глубины (z-буфер), для определения перекрываемости точек поверхностей для корректного отображения.

Пиксельный шейдер возвращает в качестве цвета пикселя номер излучающей поверхности к которой принадлежит. Так как он вызывается уже после проверки буфера глубины, то это значит, что точка видна из камеры (источника) и на неё попадает излучение, равное излучению источника деленной на полное число пикселей в центральной круговой части изображения.

Полученное изображение обрисовывается не на экран, а в текстуру, и подсчитывается количества пикселей принадлежащих каждой попавшей в кадр поверхности, и делению этой величины на полное число пикселей в круговой части изображения. Полученные числа аддитивно вносятся в таблицу угловых коэффициентов излучения для дальнейшей обработки[2-5].

Для подсчёта угловых коэффициентов на основе изображения выгодно так же использовать ресурсы графического процессора, чтобы, во-первых, не тратить время на выгрузку каждой картинке из видеопамати в память компьютера (разрешение изображения оптимально делать не менее 2000 на 2000 пикселе), во-вторых, графическое ядро справится с задачей обработки изображения значительно быстрее. Для решения поставленной задачи предлагается использовать новшество DirectX 11 – вычислительный шейдер(ComputeShader). Вычислительный шейдер это программа, заменяющая собой весь графический конвейер, позволяя использовать графическое ядро для общих задач. В данной задаче, вычислительный шейдер обрабатывает полученное изображение, и возвращает массив элементов системы, на которые излучает исследуемая излучающая поверхность и количество пикселей попавших в кадр.

### Верификация алгоритма

Для проверки работоспособности предложенного алгоритма применим его для расчёта теплового поля вакуумной печи по модели[5]. В качестве реального прототипа принята купольная вертикальная вакуумная электрическая печь сопротивления фирмы ALDVACUUMTechnologies с водяным охлаждением кожуха и тремя зонами нагрева для отжига труб. Габариты рабочего пространства печи: диаметр ~0,65 м, высота 5 м.

При проведение штатной тарировки термического оборудования с тиристорных блоков управления нагревателей снимаются показания подаваемой мощности, которые подаются на математическую модель. Сравниваются показания со штатных управляющих термопреобразователей реальной печи и модели по трем зонам нагрева: верхней, средней и нижней.

Как видно из рисунков 5-7 теоретическая модель в достаточной мере соответствует реальной. Оптические параметры системы, которые в общем случае зависят от температуры, приняты статическими на всем временном промежутке, что приводит к расхождениям при низких температурах, и практически сведению разницы к нулю при достижении уставки в 600 градусов Цельсия.

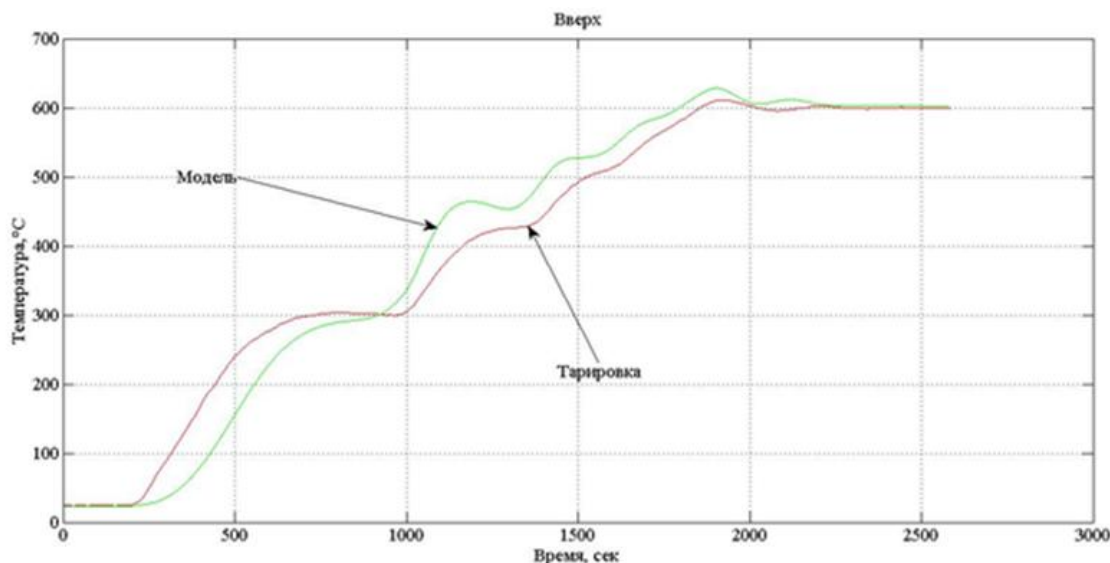


рис. 5 Сравнение показаний по верхней зоне нагрева

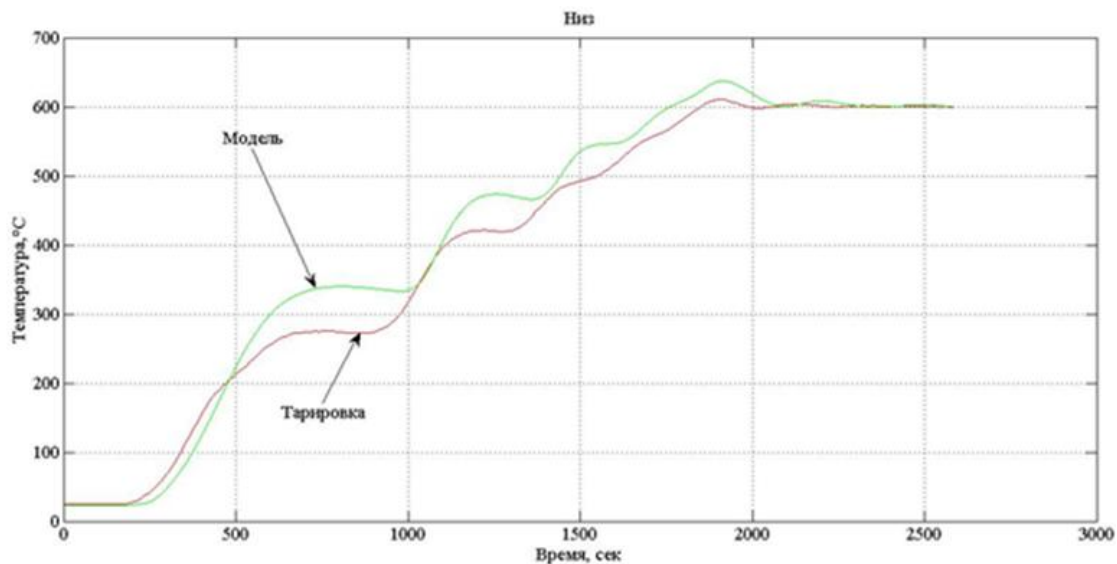


рис. 6 Сравнение показаний по нижней зоне нагрева

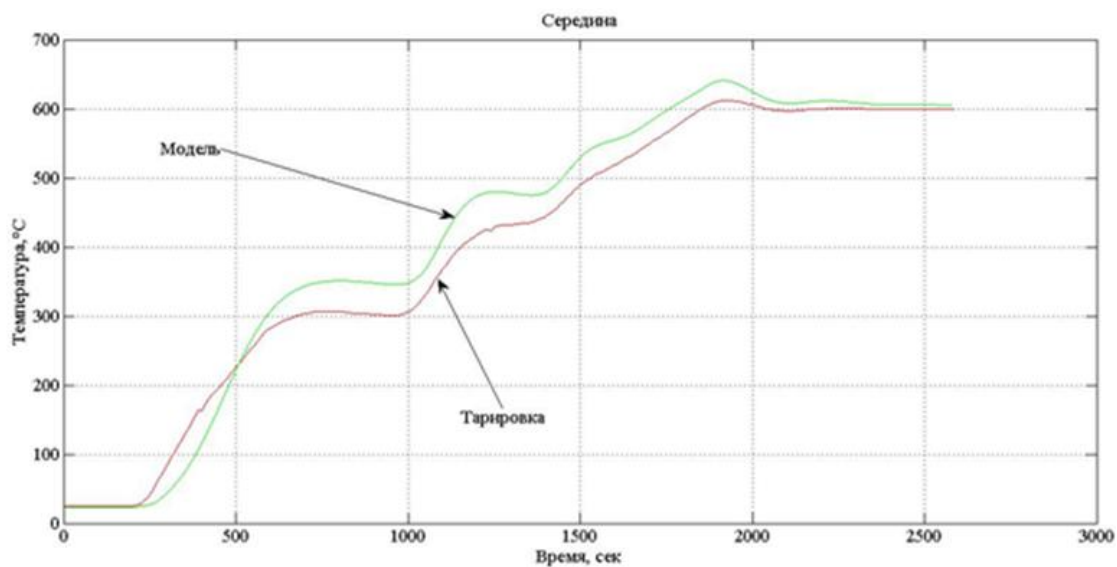


рис. 7 Сравнение показаний по нижней зоне нагрева

### Вывод

Предложен принципиально новый способ расчета угловых коэффициентов излучения основанный на использовании ресурсов графического процессора, Разработано программное обеспечение и проведены экспериментальные исследования. Данный способ позволяет рассчитывать угловые коэффициенты в системе произвольной сложности с любой необходимой точностью. Теоретическая модель в достаточной мере соответствует реальной, что позволяет использовать её для анализа процессов теплопередачи.

### Литература

1. Лебедев И.А. Использование графического конвейера для расчета угловых коэффициентов излучения // XV Международная конференция «Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта(CAD/CAM/PDM - 2015)»: сб. трудов. – Москва: ИПУ РАН, 2015. – С. 187-190
2. Лебедев И.А. Модель термообработки металлических изделий в вакууме / И.А. Лебедев, А.П. Пономарев // VII Международная научно-практическая конференция «Энергосберегающие технологии в промышленности. Печные агрегаты. Экология»: сб. материалов. – Москва: МИСиС, 2014. – С. 270-280.
3. Лебедев И.А. Управление равномерностью нагрева изделий в вакуумных электрических печных агрегатах / Лебедев И.А., Шатов Д.В.// XII Всероссийское совещание по проблемам управления: сб. трудов. – Москва: ИПУ РАН, 2014. – С. 4315-4324.
4. Лебедев И.А. Компьютерная модель нагрева теплозащитного контейнера в вакуумных печах / Лебедев И.А., Пономарев А.П. // VI Международная научно-практическая конференция «Печные агрегаты и энергосберегающие технологии в промышленности. Экология»: сб. трудов. – Москва: МИСиС, 2012. – С. 319-320.
5. Лебедев И.А. Модель термообработки металлических изделий в вакуумных электрических печах сопротивления/ И.А.Лебедев, А.П.Пономарев//Доклады Томского государственного технологического университета систем управления и радиоэлектроники, Том.19, № 1. -Томск, 2016. -С.83-89.