

¹ Нижегородский государственный архитектурно-строительный университет, г. Нижний Новгород

² Волжский государственный университет водного транспорта, г. Нижний Новгород

АВТОМАТИЗАЦИЯ МОДЕЛИРОВАНИЯ ПОВЕРХНОСТИ СУДОВОГО КОРПУСА

В статье представлен модифицированный алгоритм автоматизированного формирования геометрической модели поверхности судового корпуса. Показано, что использование разработанного алгоритма построения NURBS-поверхностей позволяет в несколько раз снизить трудоемкость проведения работ по сравнению с использованием ручного труда при моделировании поверхностей в стандартных судостроительных системах моделирования типа SeaSolution, Rhino и других. Кроме того, актуальность данного исследования существенно возрастает в условиях зарубежного санкционного давления в плане импортозамещения. На основе данного алгоритма создано программное приложение, реализованное с использованием языков HTML5, JavaScript. Приложение позволяет решать задачи создания и отображения трехмерных моделей судовых поверхностей в стандартном интернет-браузере. Программа позволяет оперативно анализировать варианты геометрии модели корпуса судна путем корректировки исходных данных, что существенно повышает производительность труда конструктора. Предусмотрен экспорт сформированной NURBS-поверхности для использования в других системах проектирования, в том числе для моделирования конструкций корпуса судна, проведения расчетов остойчивости, раскроя обшивки и решения других задач.

Ключевые слова: кривая NURBS, поверхность NURBS, поверхность корпуса судна, геометрическая модель, языки программирования JavaScript, HTML5.

Введение. Внедрение полного цикла *CALS/ИПИ (Информационная Поддержка жизненного цикла Изделия)* — технологии в судостроении является одной из главных задач отрасли [1–4]. Специалисты конструкторских бюро ведущих предприятий отмечают, что конкурентоспособное проектирование начинается с моделирования электронной модели судового корпуса. Выверенная геометрическая электронная модель обеспечивает дальнейшее качество работы с внутренним насыщением корпуса и с конструированием внешней оснастки [5, 6].

Компьютерное моделирование поверхности вручную зачастую приводит к ошибкам сглаживания, что влечет за собой критические затруднения при гибке и сварке листов обшивки [7]. Важно выполнять работу по созданию геометрической поверхности, обладающей монотонной гладкостью, по крайней мере, второго порядка. При этом все изменения должны быть интерактивными, в реальном режиме и легко доступными для пользователя [8]. Выглаженная поверхность позволяет достигать качественного построения разверток листов обшивки [9].

Проблема создания качественной поверхности корпуса корабля имеет большое значение при ав-

томатизированном проектировании судов. Гладкость второго порядка этих поверхностей является одним из важнейших условий точности изготовления конструкций корпуса корабля. Это приводит к экономии времени и повышению качества сборки и сварки корпуса. А также значительно снижает стоимость конструкции. В настоящее время используется несколько методов контроля качества поверхности судов:

1. Визуализация гауссовой кривизны, позволяющая выявить проблемные участки поверхности корпуса корабля.
2. Визуализация радиусов кривизны кривых, поверхностей и сечений.
3. Визуализация точек перегиба и линий перегиба сечений и поверхностей.
4. Динамическое изменение линий перегиба и кривизны при ручном редактировании поверхности.
5. Визуализация поверхностей и сечений в сжатом виде по одной из осей координат. Это очень важная функция для моделирования поверхностей, сильно вытянутых по одной из осей координат (крылья, руль и т. д.).
6. Автоматический контроль отклонения поверхности от исходных данных.

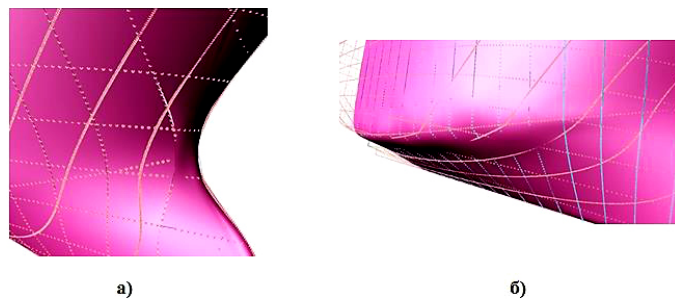


Рис. 1. Неточности алгоритма:
 а) разрыв координат и производных в месте стыковки носовой и срединной поверхности; б) неточность моделирования скулового слoma

Описанные средства контроля качества поверхности позволяют практически полностью отказаться от бумажных чертежей (или, по крайней мере, свести их к минимуму), а также существенно сократить время моделирования и повысить качество моделируемой поверхности [10]. При проектировании поверхности корпуса корабля с помощью судостроительных CAD-систем основной целью является сокращение ручного труда. Преимущества использования полностью или частично автоматизированного построения поверхности корпуса судна совершенно очевидны. Для проектировщика высокой квалификации качественная работа по формированию гладкой, монотонной поверхности в стандартных системах геометрического моделирования вручную занимает примерно 40 часов. Как правило, это зависит от возможностей графической программы, в которой реализуется проект, и от опыта специалиста. С целью снижения временных затрат в статье [11] был представлен автоматизированный алгоритм построения электронной геометрической модели поверхности судового корпуса. При анализе полученных результатов было выявлено неточное моделирование поверхности в области острой скулы корпуса судна, а также примыкание носовой оконечности и средней части корпуса.

Данная работа посвящена описанию модифицированного алгоритма построения судовой поверхности, позволяющего избежать неточностей построения, упомянутых выше.

Описание алгоритма. Одним из основных требований к разработанному алгоритму являлось максимально близкое прилегание поверхности к заданному массиву точек по всей площади корпуса, в том числе и в зонах, прилегающих к носовой оконечности произвольной формы, а также зонам интенсивного изменения геометрии. Обычно такими зонами являются острые скуловые линии корпусов некоторых типов судов.

Анализ неточностей алгоритма (рис. 1), описанного в работе [11], показал, что зачастую появляющиеся на этапе построения поверхности ее несовершенства могут быть следствием невысокого качества теоретического чертежа (ТЧ), являющегося источником исходных данных для построения. Низкое качество ТЧ может проявляться в плохой согласованности его основных линий, шпангоутов, батоксов, ватерлиний, что влечет появление бухтин, вмятин на поверхности и ее волнистость, это во-первых. Во-вторых, раздельное формирование поверхностей в разных зонах корпуса разными методами может привести к появлению разрывов поверхности и ее производных на границах

стыковки частей поверхности (рис. 1а). В-третьих, в районах особенностей типа скуловых сломов поверхность плохо отслеживает реальную геометрию корпуса (рис. 1б).

Модифицированный алгоритм состоит из следующих шагов:

1. Как отмечалось в работе [11], исходной информацией для алгоритма являются линии шпангоутов, батоксов и ватерлиний ТЧ в формате DXF. Эти линии размещаются в трехмерном пространстве на своих числовых отметках для формирования каркасной модели корпуса судна. Затем средствами CAD системы (в большинстве случаев это AutoCAD) на основе этих линий генерируется облако точек, причем оптимальным для большинства конструкций корпуса судна является количество точек 300–400 по линиям батоксов и ватерлиний по их длине, и 60–90 — для линий шпангоутов.

2. Опытная эксплуатация алгоритма показала, что в большинстве случаев точки, расположенные на батоксах (за исключением диаметрального батокса) практически не влияют на качество и точность построения поверхности. Поэтому в дальнейшем эти точки рекомендуется исключить из модели во избежание их отрицательного влияния на вычислительный процесс. Кроме того, удаляются все совпадающие друг с другом в пространстве точки, поскольку для алгоритма эта ситуация является критичной. Все манипуляции с облаком точек наиболее удобно выполнять в рамках системы КЗ [12]. Подготовленное облако точек экспортируется в текстовый файл специально созданной макрокоманды системы КЗ:

```
defarr xm[15];

fname="F:\\Ship Hull_experiment\\Models\\
Hull_2.xyz";

//_____Выбор и сохранение точек

i=0;
lab_1:
i=i+1;
If (i<=sysv) {
if (GetObjType(GetObjNum(i))=1) {
rr=GetObjGeo(GetObjNum(i),xm);
str1=Str(xm[1]);
str2=Str(-xm[2]);
str3=Str(xm[3]);
str_all=str1+" "+str3+" "+str2;
pos=0;
if (i==1) {pos=-1;}
st=PutStr(fname,str_all,pos);
```

```

}
goto lab_1;
}
exit;

```

3. Система координат, в которой следует разместить полученное облако, ориентирована следующим образом: начало координат помещается как можно ближе к корме, ось X направляется от кормы в носовую часть, ось Z — вверх. Ось Y направляется таким образом, чтобы система координат образовывала правую тройку (рис. 2). Теоретические шпангоуты нумеруются от начала координат (0-вой шп.) в направлении носовой части. При наличии шпангоутов, расположенных в корму от 0-вого шпангоута, они нумеруются в направлении кормы как -1-ый, -2-ой и т.д.

В текстовом файле последней строкой размещаются значения шести дополнительных параметров, управляющих процессом автоматического формирования поверхности:

Space — Теоретическая шпация

First — Номер шпангоута, до которого будет строиться вся поверхность

Second — Номер шпангоута, до которого будет строиться средняя часть поверхности

Third — шпангоута, до которого будет строиться носовая часть поверхности

Stern — Параметр наличия ахтерштевня (0 — нет, 1 — есть)

Step — Шаг выбора точек на теоретическом шпангоуте (см. ниже)

4. Построение главных линий поверхности производится путем формирования 3D NURBS-кривых с использованием библиотеки *Verb Nurbs* [14]. Первой линией является линия диаметрального батокса. Для ее построения из всего облака точек выделяется подмножество точек, координаты которых по оси Y = 0. В подмножестве отыскивается точка, расположенная в верхней части носовой поверхности корпуса путем сравнения координат точек по осям X и Z, эта точка помещается в начале списка точек в подмножестве. Остальные точки выстраиваются в направлении уменьшения значений координат по осям X и Z. Сформированный список используется для построения NURBS-кривой с помощью функции *verb.geom.NurbsCurve.byPoints*. Линия диаметрального батокса (ЛДБ) строится до самой последней точки, которая может быть найдена в списке обладающих координатой Y = 0.

5. Построение линии, соответствующей линии планширя фальшборта (ЛПФ), если он есть, на ТЧ. Если же фальшборт не входит в ТЧ, далее строится линия примыкания палубы к наружной обшивке. Первым действием на этом шаге является включение в качестве первой точки в новое подмножество точки, которая является первой в подмножестве, используемом на шаге 4 (см. выше) для построения

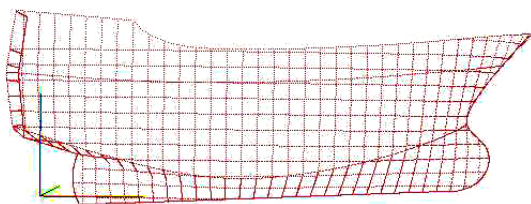


Рис. 2. Положение системы координат

ЛДБ. Все последующие точки выбираются из общего множества, исходя из нескольких условий

$$\begin{cases}
 p_{i+1} = \min(\overline{dis}(p_i, p_j)), \\
 \overline{dis}(p_i, p_2), \overline{OY}) \neq 0, \\
 -0,11 \leq \overline{dis}(p_i, p_{i+1}), \overline{OZ} \leq 0,73, \\
 \overline{dis}(p_i, p_{i+1}), \overline{dis}(p_{i-1}, p_i) \leq -0,89,
 \end{cases} \quad (1)$$

где $\overline{dis}(p_i, p_{i+1})$ — нормализованный вектор между точками p_i и p_{i+1} ;

i — номер последней добавленной в подмножество точки;

j — номер следующей искомой точки.

Первое условие в списке (1) означает, что следующая искомая точка в последовательности должна быть ближайшей к последней уже добавленной. Второе условие должно быть проверено при поиске второй точки в последовательности. Вектор от первой точки до второй не должен быть перпендикулярен оси координат Y для исключения из рассмотрения точек, принадлежащих ЛДБ. Третье условие требует, чтобы нормализованный вектор между точками p_i и p_{i+1} лежал бы в пределах от 43° до 97° по отношению к оси Z. Четвертое условие означает, что угол между векторами $\overline{dis}(p_i, p_{i+1})$ и $\overline{dis}(p_{i-1}, p_i)$ должен быть не меньше, чем 155° , что гарантирует исключение из рассмотрения уже выбранных точек. Все численные значения углов в условиях (1) выбраны путем тщательного анализа самых разнообразных форм ЛПФ. Поскольку добавление точек в список ведется последовательно с носа в корму, процесс заканчивается, как только произведен выбор точки, координата которой по оси X становится меньше, чем координата по этой же оси теоретического шпангоута, определяемого параметром *First*. Для построения линии также используется функция *verb.geom.NurbsCurve.byPoints*.

6. Третьей, и заключительной линией является линия, соответствующая линии теоретического шпангоута (ЛТШ), определяемая двумя параметрами *Space* и *Third*. Эта линия необходима для раздельного построения носовой и других поверхностей корпуса. Дело в том, что носовая поверхность в большинстве случаев имеет сложную конфигурацию и требует специфических подходов для формирования. Формирование списка сводится к простому выбору точек, координаты которых по оси X соответствуют координатам точек шпангоута. Точки, необходимые для построения линии, лежащие непосредственно на ЛДБ и ЛПФ, отыскиваются с использованием функции *verb.geom.NurbsCurve.closestPoint* для каждой линии и помещаются в начало и в конец списка точек, по которым строится ЛТШ с помощью функции *verb.geom.NurbsCurve.byPoints*.

В этих же точках производится разрыв ЛДБ и ЛПФ для отдельного использования полученных ветвей в построении носовой и срединной поверхностей корпуса судна (рис. 3).

7. Построение поверхности носовой оконечности. Ниже, в пункте 8, описана процедура построения поверхности корпуса в средней части судна на базе линий теоретических шпангоутов. Однако в носовой оконечности данный подход неприменим, поскольку в большинстве случаев носовая оконечность имеет сложную форму, которая теоретические шпангоуты в этой части разрывает на две ветви (рис. 3). В этом случае для построе-

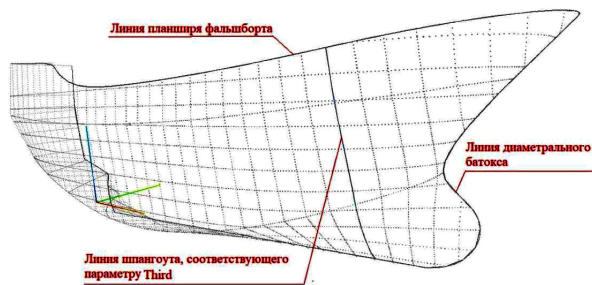


Рис. 3. Главные линии корпуса судна

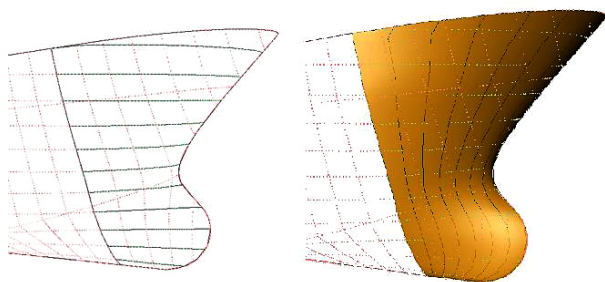


Рис. 4. Построение NURBS-поверхности в носовой части

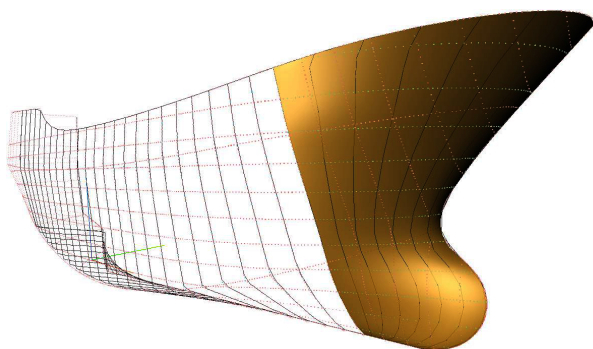


Рис. 5. NURBS-кривые для построения поверхности в средней части

ния поверхности используются точки, принадлежащие ватерлиниям в носовой части. Точки выделяются последовательно для каждой ватерлинии с помощью движущегося снизу вверх узкого сепаратора шириной $Z_{max}/5000$, где Z_{max} — максимальная координата по оси Z точки стыковки двух линий ЛДБ и ЛТШ. По выделенным точкам строятся NURBS-кривые, изображённые на рис. 4. Список линий дополняется двумя ветвями линий ЛДБ и ЛТШ, «подлеченными» на шаге 6 путем разрыва этих линий. Данный список линий используется для построения поверхности с помощью функции *verb.geom.NurbsCurve.byPoints*. Поскольку, как показано ниже, алгоритм получения поверхности в средней части и в носу отличаются, произведено переопределение полученной поверхности кривыми другого направления. Для этого к полученной поверхности применена функция *verb.geom.NurbsCurve.isocurve*, позволяющая получить NURBS-кривые по одному из заданных параметрических координат поверхности. Целесообразно количество кривых варьировать от 6 до 8. Данные кривые добавляются в список всех кривых для построения итоговой поверхности.

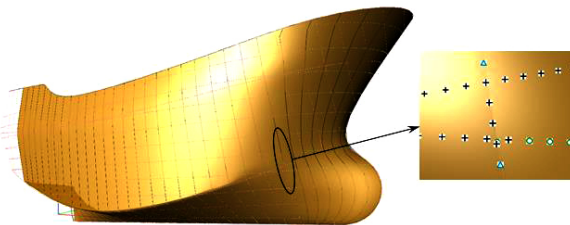


Рис. 6. Разрыв в первых производных NURBS-поверхности носовой и средней поверхностей

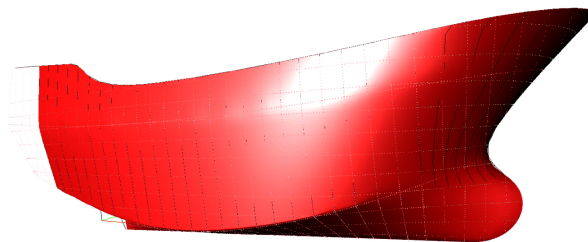


Рис. 7. Объединенная NURBS-поверхность корпуса

8. Построение поверхности в средней части корпуса. В этой части поверхность строится по NURBS-кривым теоретических шпангоутов, для построения которых выбираются из общего списка точки, имеющие координаты соответствующего сечения корпуса от X. Эти координаты вычисляются с использованием номеров шпангоутов в диапазоне между параметрами *Second* и *Third* и параметра *Space*. Для построения кривых теоретических шпангоутов используются не все точки, которые удается выделить для данного шпангоута из общего списка точек. Точки выбираются из списка точек *i*-го шпангоута с помощью заданного параметра *Step*. Это, с одной стороны, уменьшает количество точек для проведения вычислений, с другой стороны, как видно из рис. 16, у ряда судов имеется острый скуловой слом, который необходимо учитывать при построении кривых. С целью выделения участков кривых в районе скуловых сломов производится проверка скалярного произведения двух векторов между узлами i и $i+1$, а также i и $i-1$ с проверкой условия $(\text{dis}(p_i, p_{i+1}) \text{dis}(p_{i-1}, p_i)) \leq -0,97$, что соответствует углу между этими векторами 165° . Если условие выполняется, то точки $i+1$ и $i-1$ также добавляются в общий список для построения кривых. Результат построения кривых с помощью функции представлен на рис. 5.

9. Заключительным шагом является построение итоговой поверхности. Поверхности в средней и носовой части, изображенные на рис. 6, являются NURBS-поверхностями в нашем случае 3-го порядка, непрерывными между собой по значениям координат, однако имеющие разрывы в производных начиная с первого порядка в месте стыковки в районе шпангоута, определяемого параметром *Third*.

Для обеспечения гладкости по всем производным необходимо построить итоговую поверхность от шпангоута *Second* до носовой части. Для этого используется полный список всех кривых от изолиний, найденных на шаге 7, до кривых, использо-

ванных на шаге 8. Определяющим для построения объединённой поверхности на этом шаге является параметр *Stern*, принимающий значение 0 или 1. *Stern* = 1 означает, что в кормовой оконечности есть килевая часть, оканчивающаяся ахтерштевнем, в этом случае параметр *Second* должен быть равен номеру шпангоута, последнему перед ахтерштевнем (рис. 7). В этом случае часть поверхности до кормового транца остается недостроенной. В случае, если параметр *Stern* = 0, достаточно задать *Second* = *First*, при этом поверхность строится вплоть до транца, то есть до шпангоута номер *First*.

10. Особый случай представляет собой *Stern* = 1. В этом случае участок кормовой поверхности между шпангоутами *First* и *Second* приходится достраивать отдельно. Для обеспечения неразрывности поверхностей в средней части и кормовой по координатам шпангоут номер *Second* необходимо разорвать в месте стыка кормового киля и днища (рис. 8) и полученную ветвь использовать как первую кривую для построения поверхности в корме. Остальные кривые строятся в этой части аналогично тому, как описано в пункте 8. По данным кривым строится недостающая часть кормовой поверхности. При этом, однако, наблюдается разрыв по производным начиная с первого порядка. Преодоление данного недостатка предполагается осуществить в последующих версиях алгоритма.

В случае *Stern* = 0 данный пункт алгоритма выполнять не требуется.

Программная реализация алгоритма. Алгоритм, описанный выше, реализован в виде макета программного продукта, предназначенного для автоматизированного построения поверхности корпуса судна по облаку точек, полученному на базе теоретического чертежа. Программный продукт реализован для всех типов персональных ЭВМ как платформо-независимый благодаря использованию языков программирования JavaScript, HTML и CSS. Для реализации программы требуется только наличие стандартного интернет-браузера, поддерживающего библиотеку *WebGL*. Тип и версия операционной системы: Windows 7/8/10/11, размер программы 4,4 Мб.

Для получения входных данных программы необходимо исходный теоретический чертеж корпуса, представленный в формате DXF, преобразовать в трехмерную проволочную модель, а затем на основе линий модели сгенерировать облако точек, элементы которого совпадают с главными линиями модели (ватерлиниями, батоксами и шпангоутами) с определенным шагом вдоль этих линий. Координаты точек затем с помощью специально созданного макроса на языке макропрограммирования системы КЗ экспортируются в текстовый файл, который является файлом исходных данных программы. Программа на основе полученных исходных данных первоначально полностью автоматически строит главные линии корпуса, а именно линию диаметрального батокса и либо линию стыка палубы и борта судна, либо линию планширя фальшборта, если он присутствует на чертеже. После этого корпус делится на носовой регион, регион средней части корпуса и кормовой регион. Автоматически строятся ватерлинии в носовом регионе, по которым далее генерируется NURBS-поверхность в носу. В средней и кормовой частях автоматически строятся NURBS-линии шпангоутов, по которым далее генерируются NURBS-поверхности в этих регионах. Поверхности в носу и в средней части затем

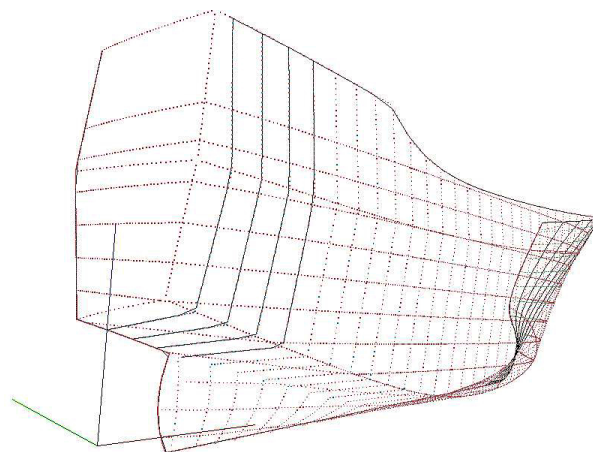


Рис. 8. NURBS-кривые для построения кормовой поверхности при *Stern* = 1

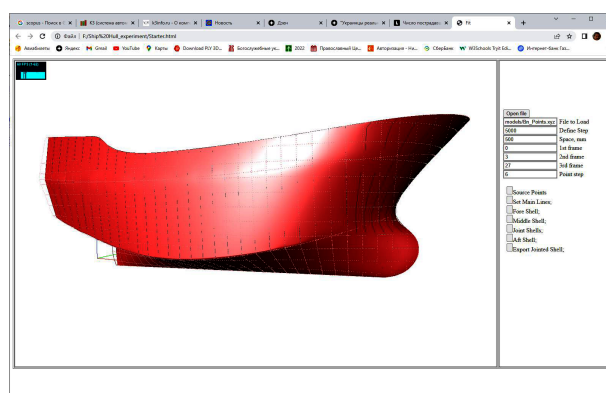
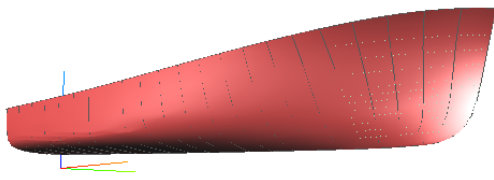
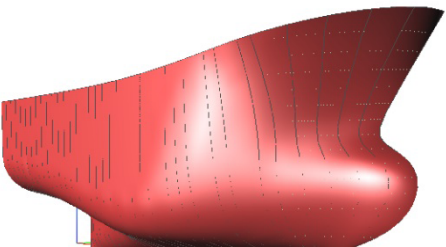
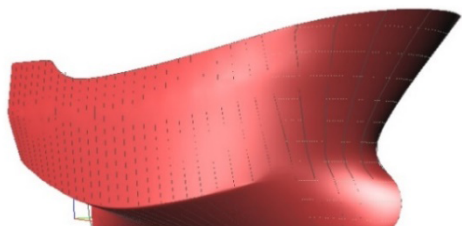
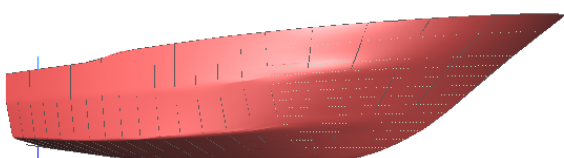
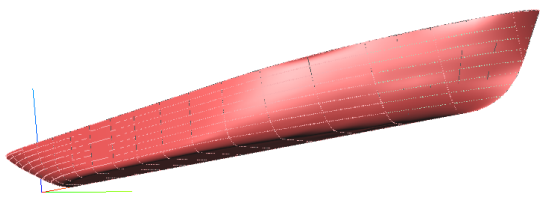


Рис. 9. Пользовательский интерфейс программы

объединяются в одну поверхность, которая далее может быть экспортирована в отдельный файл для использования в CAD-системах проектирования судостроительных конструкций. Имеется также возможность для корпусов некоторых типов в носовой, средней и кормовой части генерировать поверхность как одно целое. Программный продукт осуществляет экспорт сформированной поверхности.

Визуализация всех геометрических элементов, включая поверхности, линии и точки, осуществляется при помощи библиотеки *Three.js* [13], которая обеспечивает удобный интерфейс к функциям библиотеки *WebGL*. Задание кривых произвольной формы возможно двумя способами: указание набора точек и интерполяция гладкой кривой между ними или выбор базовых контрольных точек кривой определенной степени. В описанном выше алгоритме используется интерполяционная разновидность NURBS-кривых и поверхностей, поскольку форма, которую должна принять линия, известна, а данный тип кривых и поверхностей обеспечивает отсутствие погрешности построения от заданной геометрии. Используемая для моделирования кривых и поверхностей библиотека *Verb Nurbs* [14] представляет собой базовый класс NURBS-геометрии.

Пользовательский интерфейс программы является дружелюбным и наглядным. Основные функции, которые могут быть активированы интерфейсом, полностью соответствуют шагам алгоритма, описанного выше, от загрузки с диска исходных

№	Модель	Изображение	Время построения
1	Разъездной катер проекта 371 (12,5 м)		24 сек
2	Рыболовное судно вп 27 (18 м)		38 сек
3	Рыболовное судно вп 10 (22 м)		36 сек
4	Моторная яхта «Дон-20» (22 м)		30 сек
5	Речной сухогруз (96 м)		27 сек

данных до выгрузки информации о сформированной поверхности (рис. 9).

С помощью разработанного программного продукта было выполнено моделирование поверхностей судов нескольких проектов различного типа (табл. 1). Как показали результаты моделирования, была достигнута высокая точность полученных поверхностей, отклонение от облака точек нигде не превышало 0,001 от полуширины корпуса при сохранении гладкости поверхности до вторых производных. Кроме того, трудоемкость построения поверхностей снижена до 40 раз по сравнению с традиционными технологиями построения судовых поверхностей с помощью известных систем моделирования. Моделирование выполнено на процессоре Intel(R) Core(TM) i5-5200U CPU @.

Выводы и заключения. Общее время построения поверхности корпуса судна варьируется в пределах 40 секунд. Время генерации корпуса увеличивает-

ся в зависимости от вида носовой оконечности, чем сложнее форма, тем дольше скорость обработки.

Использованная для моделирования кривых и поверхностей библиотека *Verb Nurbs* предоставляет широкий спектр функций сплайновой геометрии. Преимуществом ее является то, что она реализована на нескольких языках программирования высокого уровня, включая C#, Python, JavaScript, Java. Это позволяет внедрять описанный алгоритм в другие системы моделирования поверхностей судового корпуса.

Разработанный макет программы предназначен для решения задачи создания и отображения трехмерных моделей судовых поверхностей в стандартном интернет-браузере. Программа существенно повышает производительность труда конструктора.

В настоящей статье описан алгоритм построения непрерывной поверхности судового корпуса вплоть до кормового транца. Алгоритм позволяет обеспе-

читать максимально близкое прилегание поверхности к заданному массиву точек по всей площади корпуса, в том числе в зонах интенсивного изменения геометрии.

Дальнейшее совершенствование алгоритма предполагает автоматизацию построения непрерывной поверхности корпуса по всей его длине.

Библиографический список

1. Доросинский Л., Зверева О. CALS технологии. LAP LAMBERT Academic Publishing, 2014. 277 с. ISBN 978-3-659-67118-0.

2. Аведьян А. PLM 2.0 в судостроении. URL: <https://www.informdom.com/metalloobrabotka/2010/3/plm-20-v-sudostroenii.html> (дата обращения: 02.05.2021).

3. Ковшов А. Н., Назаров Ю. Ф., Ибрагимов И. М. [и др.]. Информационная поддержка жизненного цикла изделий машиностроения: принципы, системы и технологии CALS/ИПИ. Москва: Издат. центр Академия, 2007. 304 с.

4. Сулов А. Н., Одегова О. В. Внедрение CALS-технологий в судостроении. URL: http://old.cals.ru/conferences/archive/CALS/cals_2001/tez/spgmtu.pdf (дата обращения: 02.05.2021).

5. Войткунский Я. И., Фаддеев М. И., Федяевский К. К. Гидромеханика. 2 изд. Ленинград: Судостроение, 1982. 472 с.

6. Гажиев А. В., Кошкалда Н. В. Судостроительное черчение. Ленинград: Судостроение, 1979. 184 с.

7. Алексанов А. Некоторые факты о моделировании судовых поверхностей, о которых вам не расскажут. URL: <https://ru.shmexpert.com/feed/archive/2022/11> (дата обращения: 20.01.2023).

8. Косников Ю. Н. Развитие теории геометрического моделирования пространственных форм и совершенствование графических систем реального времени: дис. д-ра техн. наук. Пенза: Изд-во ПГУ, 2006. 373 с.

9. Shorkina I., Novikov S. Development of the skin of the surface of the tunnel part of the ship's hull // CEUR Workshop Proceedings. 31. Сер. «GraphiCon 2021 – Proceedings of the 31st International Conference on Computer Graphics and Vision». 2021. С. 1098–1103.

10. Popov E. V., Rotkov S. I. The retrieval of NURBS-surface by genetic algorithm on the basis of point cloud // 21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG) Communication Papers Proceedings. Plzen, Czech, June, 2013.

11. Попов Е. В., Шоркина И. Н. Автоматическое построение NURBS поверхности корпуса корабля // GraphiCon

2022: тр. 32-й Междунар. конф. по компьютерной графике и машинному зрению (Рязань, 19–22 сент. 2022 г.). Москва: Ин-т прикладной математики им. М. В. Келдыша РАН, 2022. С. 162–169. DOI: 10.20948/graphicon-2022-162-169.

12. Система трехмерного моделирования K3. URL: <http://www.k3info.ru/> (дата обращения: 30.04.2023).

13. Хавербеке М. Выразительный JavaScript. Современное веб-программирование. Санкт-Петербург: Питер, 2022. 480 с.

14. Verb библиотека. URL: <http://verbnurbs.com/docs/> (дата обращения: 25.04.2023).

ПОПОВ Евгений Владимирович, доктор технических наук, профессор кафедры «Инженерная геометрия, компьютерная графика и автоматизированное проектирование» Нижегородского государственного архитектурно-строительного университета (ННГАСУ), г. Нижний Новгород.

SPIN-код: 8156-5998

AuthorID (РИНЦ): 663542

ORCID: 0000-0002-3058-2369

AuthorID (SCOPUS): 56585437200

ResearcherID: F-6001-2013

Адрес для переписки: porov_eugene@list.ru

ШОРКИНА Ирина Николаевна, аспирант кафедры «Инженерная геометрия, компьютерная графика и автоматизированное проектирование» ННГАСУ, г. Нижний Новгород; доцент кафедры «Теория конструирования инженерных сооружений» Волжского государственного университета водного транспорта, г. Нижний Новгород.

SPIN-код: 2754-3332

AuthorID: 826342

ORCID: 0000-0002-0960-1801

ResearcherID: IAO-8863-2023

Адрес для переписки: irenika77@gmail.com

Для цитирования

Попов Е. В., Шоркина И. Н. Автоматизация моделирования поверхности судового корпуса // Омский научный вестник. 2023. № 4 (188). С. 13–21. DOI: 10.25206/1813-8225-2023-188-13-21.

Статья поступила в редакцию 25.05.2023 г.

© Е. В. Попов, И. Н. Шоркина

¹ Nizhegorodsky State
Architectural and Civil
Engineering University,
Nizhny Novgorod, Russia

² Volga State University
of Water Transport,
Nizhny Novgorod, Russia

AUTOMATION OF SHIP HULL SURFACE SIMULATION

The article presents a modified algorithm for automated formation of geometric model of ship hull surface. The use of developed algorithm for construction of NURBS-surfaces allows to decrease labor input in several times in comparison with the use of manual labor when modeling surfaces in standard shipbuilding modeling systems like Sea Solution, Rhino and others. On the basis of this algorithm a software application is created, implemented in HTML5, JavaScript. The application allows to solve problems of creating and displaying three-dimensional models of ship surfaces in a standard the Internet browser. The program allows you to quickly analyze variants of the geometry of the ship hull model by adjusting the initial data, which significantly increases the productivity of the designer. It is possible to export the formed NURBS-surface for use in other design systems, including ship hull construction modeling, stability calculations, hull shell cutting and other tasks.

Keywords: NURBS curve, NURBS-surface, hull surface, geometric model, programming languages: JavaScript, HTML5.

References

1. Dorosinskiy L., Zvereva O. CALS tekhnologii [CALS technologies]. LAP LAMBERT Academic Publishing, 2014. 277 p. ISBN 978-3-659-67118-0. (In Russ.).
2. Aved'yan A. PLM 2.0 v sudostroyenii [PLM 2.0 in shipbuilding]. URL: <https://www.informdom.com/metalloobrabotka/2010/3/plm-20-v-sudostroenii.html> (accessed: 02.05.2021). (In Russ.).
3. Kovshov A. N., Nazarov Yu. F., Ibragimov I. M. [et al.]. Informatsionnaya podderzhka zhiznennogo tsikla izdeliy mashinostroyeniya: printsipy, sistemy i tekhnologii CALS/IPI [Information support for the life cycle of engineering products: principles, systems and technologies of CALS/IPI]. Moscow, 2007. 304 p. (In Russ.).
4. Suslov A. N., Odegova O. V. Vnedreniye CALS-tekhnologii v sudostroyenii. [Implementation of CALS-technologies in shipbuilding]. URL: http://old.cals.ru/conferences/archive/CALS/cals_2001/tez/spgmtu.pdf (accessed: on 02.05.2021). (In Russ.).
5. Voytkunskiy Ya. I., Faddeyev M. I., Fedyayevskiy K. K. Gidromekhanika. [Hydromechanics]. Leningrad, 1982. 472 p. (In Russ.).
6. Gazhiyev A. V., Koshkalda N. V. Sudostroitel'noye chereniye. [Shipbuilding drafting]. Leningrad, 1979. 184 p. (In Russ.).
7. Aleksanov A. Nekotoryye fakty o modelirovanii sudovoy poverkhnosti, o kotorykh vam ne rasskazhut [Some facts about ship surface modeling that you won't be told]. URL: <https://ru.shmexpert.com/feed/archive/2022/11> (accessed: 20.01.2023). (In Russ.).
8. Kosnikov Yu. N. Razvitiye teorii geometricheskogo modelirovaniya prostranstvennykh form i sovershenstvovaniye graficheskikh sistem real'nogo vremeni [Development of the

theory of geometric modeling of spatial shapes and improvement of real-time graphic systems]. Penza, 2006. 373 p. (In Russ.).

9. Shorkina I., Novikov S. Development of the skin of the surface of the tunnel part of the ship's hull // CEUR Workshop Proceedings. 31. Ser. «GraphiCon 2021 – Proceedings of the 31st International Conference on Computer Graphics and Vision». 2021. P. 1098–1103. (In Engl.).

10. Popov E. V., Rotkov S. I. The retrieval of NURBS-surface by genetic algorithm on the basis of point cloud // 21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG) Communication Papers Proceedings. Plzen, Czech, June, 2013. (In Engl.).

11. Popov E. V., Shorkina I. N. Avtomaticheskoye postroyeniye NURBS poverkhnosti korpusa korablya [Automatic construction of the NURBS surface of the ship's hull] // GraphiCon 2022. Moscow, 2022. P. 162–169. DOI: 10.20948/graphicon-2022-162-169. (In Russ.).

12. Sistema trekhmernogo modelirovaniya K3 [Three-dimensional modeling system K3]. URL: <http://www.k3info.ru/> (accessed: 30.04.2023). (In Russ.).

13. Khaverbeke Mareyn. Vyzritel'nyy JavaScript. Sovremennoye veb-programirovaniye [Expressive JavaScript. Modern Web Programming]. Saint Petersburg, 2022. 480 p. (In Russ.).

14. Verb biblioteka [Verb library]. URL: <http://verbnurbs.com/docs/> (accessed: 25.04.2023). (In Russ.).

POPOV Evgeniy Vladimirovich, Doctor of Technical Sciences, Professor of Engineering Geometry, Computer Graphics and Computer-Aided Design Department, Nizhny Novgorod State University of Architecture and Civil Engineering (NNGASU), Nizhny Novgorod.
SPIN-code: 8156-5998
AuthorID (RSCI): 663542

ORCID: 0000-0002-3058-2369
AuthorID (SCOPUS): 56585437200
ResearcherID: F-6001-2013

Correspondence address: popov_eugene@list.ru

SHORKINA Irina Nikolayevna, Graduate Student of Engineering Geometry, Computer Graphics and Computer-Aided Design Department, NNGASU, Nizhny Novgorod; Associate Professor of Engineering Structures Design Theory Department, Volga State University of Water Transport, Nizhny Novgorod.

SPIN-code: 2754-3332

AuthorID (RSCI): 826342

ORCID: 0000-0002-0960-1801

ResearcherID: IAO-8863-2023

Correspondence address: irenika77@gmail.com

For citations

Popov E. V., Shorkina I. N. Automation of ship hull surface simulation // Omsk Scientific Bulletin. 2023. No. 4 (188). P. 13–21. DOI: 10.25206/1813-8225-2023-188-13-21.

Received May 25, 2023.

© **E. V. Popov, I. N. Shorkina**