

УДК 539.3

Н.В. Пустовой, В.Е. Левин, Д.А. КрасноруцкийНовосибирский государственный технический университет,
Новосибирск, Россия**АЛГОРИТМ ЧИСЛЕННОГО РЕШЕНИЯ
НЕЛИНЕЙНОЙ КРАЕВОЙ ЗАДАЧИ
ДИНАМИЧЕСКОГО ДЕФОРМИРОВАНИЯ
ТОНКОГО СТЕРЖНЯ**

В статье представлен алгоритм разработанной подпрограммы решения двухточечной краевой задачи для системы нелинейных дифференциальных уравнений первого порядка. Новый алгоритм ранее опубликованной подпрограммы KLPALG объединил в себе основные идеи подпрограммы BVFPD (DD14AD, PASVA3) и PASSIN, реализующей методику продолжения решения по параметру. Кроме того, приведены обобщенные результаты трудов авторов в задаче о нелинейном динамическом деформировании тонкого пространственного криволинейного стержня при расчете по его дифференциальной модели.

Неизвестные функции, входящие в уравнения движения, разыскиваются в дискретных точках. По методам прямого интегрирования производные по времени выражаются через текущие координаты и найденные на предыдущих шагах по времени. Первая производная по координате заменяется конечной разностью, добавляются краевые условия. Полученная система нелинейных алгебраических уравнений решается с помощью метода Ньютона с контролем длины шага из условия сходимости. Матрица Якоби этой системы имеет блочную трехдиагональную структуру, которая поддается эффективному LU-разложению. Такая декомпозиция матрицы Якоби позволяет быстро решать соответствующие системы линейных алгебраических уравнений больших размеров. Если условие сходимости метода Ньютона дает слишком маленький шаг, тогда применяется техника продолжения решения по параметру (псевдодлина дуги). После того как решена основная система нелинейных уравнений, для уточнения узловых значений вычисляемых функций применяется так называемый метод отсроченной коррекции (deferred correction method). Этот метод позволяет вычесть из получаемого решения ошибку, внесенную аппроксимацией производной по методу конечных разностей на начальном этапе численного решения. Получаемое таким образом численное решение имеет назначенную точность. Такая методика реализована в виде подпрограммы KLPALG, алгоритм которой представлен в данной статье.

Ключевые слова: нелинейная краевая задача, тонкий криволинейный стержень, дифференциальная модель, численное решение, KLPALG, BVFPD, DD14AD, PASVA3, PASSIN.

N.V. Pustovoy, V.E. Levin, D.A. Krasnorutskiy

Novosibirsk State Technical University, Novosibirsk, Russian Federation

THE NUMERICAL ALGORITHM FOR SOLVING NONLINEAR BOUNDARY PROBLEM OF THIN ROD'S DYNAMIC DEFORMATIONS

At this paper the algorithm of the subprogram for solving a two-point boundary value problem for system of the nonlinear differential equations of the first order is presented. The new algorithm of the subprogram named KLPALG united in itself the main ideas of the subprograms BVPFD (DD14AD, PASVA3) and PASSIN realizing the technique of continuation of solution by parameter. Besides, the generalized results of works of authors in a problem of nonlinear dynamic deformation of a thin spatial curvilinear rod calculated by its differential model are presented.

The unknown functions in the equations of motion are calculated at discrete mesh points. The methods of direct integration allow us to express time derivatives by the current coordinates and coordinates and velocities calculated in the previous time steps. The first derivative of coordinate is replaced by finite difference; boundary conditions are added. The obtained system of nonlinear algebraic equations is solved by Newton method with the step length control of the convergence conditions. The Jacobi matrix of this system is of the block-tridiagonal structure which lends itself to efficient LU-decomposition. This decoupling of the Jacobi matrix allows you to quickly solve the corresponding system of linear algebraic equations of the big sizes. If the condition of convergence of Newton's method gives too small step, then used the technique of continuation of the solution on the parameter a (pseudo arc-length). As soon as the system of nonlinear equations is solved, to refine the nodal values of the calculated functions we use the deferred correction method. This method subtracts from the received solution the mistakes made by the approximation derived by the method of finite differences in the initial phase of the numerical solution. Thus obtained numerical solution is of accuracy appointed by user. This method is implemented in KLPALG subroutine which algorithm is presented in this paper.

Keywords: nonlinear boundary value problem, thin curvilinear rod, differential model, numerical solution, KLPALG, BVPFD, DD14AD, PASVA3, PASSIN.

Моделирование динамических процессов в ряде прикладных задач нелинейной механики деформируемого твердого тела удается построить на базе математической модели тонкого стержня [1–3] – упругого тела, один линейный размер которого существенно больше двух других. Введением ряда гипотез [4–6] деформирование стержня, по существу, заменяется деформированием его осевой линии, проходящей через центры жесткости поперечных сечений, ортогональных к этой осевой линии. Положение точки на стержне определяется естественной координатой – длиной дуги осевой линии. В геометрически нелинейной постановке задачи такая модель стержня позволяет описать множество разнообразных пространственных форм статического и динамического равновесия. В задаче о деформировании стержня наиболее распространены два подхода: на основе метода конечных элементов

и на основе решения краевых задач для систем дифференциальных уравнений. В данной работе используется дифференциальная модель стержня [7–10]. Преимущество используемой модели стержня перед другими известными дифференциальными моделями [2, 3, 11, 12] состоит в том, что в уравнения не входит начальная кривизна осевой линии стержня, что позволяет напрямую описывать стержни с изломами осевой линии и со скачками кривизны. При расчете динамического деформирования учитывается инерция, в отличие, например, от работы [12], где динамика представляется в виде набора статических конфигураций, а форма стержня представляет собой спираль. Задача о динамическом нелинейном деформировании тонкого криволинейного стержня представляет собой краевую задачу для системы нелинейных дифференциальных уравнений в частных производных. Решение этой задачи с помощью методов прямого интегрирования [13–15] сводится к решению последовательности двухточечных нелинейных краевых задач для системы нелинейных дифференциальных уравнений первого порядка. Для решения этих задач существует ряд подпрограмм [16–18], базирующихся на разработках прошлого столетия. Решение разыскивается в дискретных точках, производные в уравнениях заменяются конечными разностями [19, 18], строится итерационный процесс по методу Ньютона-Рафсона [20], также возможны и другие подходы, например [21]. При решении задачи о динамическом деформировании тонкого стержня на определенном этапе развития дифференциальной модели [8, 10] у авторов данной статьи появилась необходимость внести определенные изменения в текст подпрограммы BVFPD [17, 18], которая хорошо подходила для решения поставленной задачи о деформировании стержня до этого момента. Кроме внесения изменений в формирование нелинейных уравнений при применении численной методики развитие дифференциальной модели стержня [8, 10] потребовало добавить в подпрограмму метод продолжения решения по параметру в виде [22, 23]. В работах [18, 22–26], а также связанных с ними описаны основные идеи, реализованные в подпрограммах BVFPD [17], PASVA3 [18], PASSIN [22], но не приводятся подробных алгоритмов, по которым можно было бы составить программу. После анализа указанных работ авторами данной статьи была разработана своя подпрограмма KLPALG [10], которая объединила в себе основные идеи указанных выше работ и позволила решать более сложные краевые задачи [8], чем

BVPFD [17], PASVA3 [18] (саму подпрограмму PASSIN [22] найти не удалось). После этого алгоритм KLPALG претерпел существенные улучшения, которые приблизили его по скорости исполнения кода к BVPFD. Этот новый алгоритм KLPALG и представлен в данной статье.

1. Математическая модель тонкого стержня

Точки недеформированной осевой линии стержня задаются радиусом-вектором

$$\bar{\mathbf{r}}(\xi) = x_k(\xi) \cdot \bar{\mathbf{i}}_k, \quad (1)$$

где ξ – произвольный параметр, связанный с естественной координатой – длиной стержня s ; $\bar{\mathbf{i}}_{1,2,3}$ – орты глобальной (неподвижной) системы координат; $x_{1,2,3}(\xi)$ – координаты точек осевой линии, здесь и далее по повторяющимся индексам ведется суммирование от 1 до 3.

Для описания ориентации поперечных сечений стержня до деформирования используется «матрица поворота» (матрица начальной геометрии) $\beta(\xi)$. Направляющие векторы главных осей инерции поперечных сечений недеформированного стержня записываются следующим образом:

$$\bar{\mathbf{e}}_j(\xi) = \beta_{jk}(\xi) \cdot \bar{\mathbf{i}}_k, \quad (2)$$

$\bar{\mathbf{e}}_3$ – направлен по касательной к осевой линии стержня, здесь и далее неуказанные индексы пробегает значения от 1 до 3.

Направляющие векторы главных осей поперечного сечения после деформации имеют следующее выражение через компоненты матрицы начальной геометрии β и матрицы поворота при деформировании λ :

$$\bar{\mathbf{e}}_j^* = \beta_{jk} \bar{\mathbf{i}}_k^* = \beta_{jk} \lambda_{kn} \bar{\mathbf{i}}_n. \quad (3)$$

Жесткостные характеристики стержня задаются следующими величинами: $EJ_{1,2}(\xi)$ – изгибные жесткости; $GJ_3(\xi)$ – жесткость на кручение; $EF(\xi)$ – жесткость на растяжение-сжатие. Жесткости также могут зависеть от текущей деформированной конфигурации – текущей кривизны осевой линии или удлинения.

Инерционные характеристики стержня: $\rho F(\xi)$ – погонная плотность; $\rho J_{1,2,3}(\xi)$ – произведение плотности и моментов инерции поперечных сечений. Если ось жесткости стержня не проходит через центры инерции поперечных сечений [27], тогда задаются координаты центра тяжести поперечного сечения относительно центра жесткости $\Delta_{1,2}(\xi)$. Вектор положения центров масс относительно центров жесткости

$$\vec{\mathbf{R}} = \Delta_1 \vec{\mathbf{e}}_1^* + \Delta_2 \vec{\mathbf{e}}_2^* = \Delta_1 \beta_{1k} \lambda_{kn} \vec{\mathbf{i}}_n + \Delta_2 \beta_{2k} \lambda_{kn} \vec{\mathbf{i}}_n = R_n \vec{\mathbf{i}}_n, \quad R_n = (\Delta_1 \beta_{1k} + \Delta_2 \beta_{2k}) \lambda_{kn}. \quad (4)$$

На стержень могут действовать внешние распределенные нагрузки и моменты. В общем случае они могут зависеть не только от текущей точки на осевой линии, но и от времени t , пространственного положения, текущей кривизны и других параметров:

$$\vec{\mathbf{q}} = \vec{\mathbf{q}}(\xi, t, \vec{\mathbf{U}}, \lambda, \dots), \quad \vec{\mathbf{m}} = \vec{\mathbf{m}}(\xi, t, \vec{\mathbf{U}}, \lambda, \dots). \quad (5)$$

В качестве внешней распределенной нагрузки может выступать, например, собственный вес, аэродинамическая нагрузка от набегающего потока воздуха [7], силы от контактного взаимодействия витков осевой линии [28], силы инерции от пробегающего внутреннего потока жидкости и др. Кроме распределенных силовых факторов на стержень могут действовать сосредоточенные силы и моменты. Сосредоточенные нагрузки на концах стержня задаются через краевые условия. Задание сосредоточенных сил и моментов в других точках приводит к многоточечной нелинейной краевой задаче, которая может быть решена в рамках численного алгоритма решения двухточечной нелинейной краевой задачи с некоторыми модификациями в алгоритме.

Мгновенные угловые ускорения и скорости поворота поперечного сечения стержня относительно главных осей $\mathbf{e}_{1,2,3}^*$ имеют следующий вид [8]:

$$\begin{aligned} \dot{\Omega}_1 &= -\ddot{\lambda}_{mn} \beta_{3m} \lambda_{kn} \beta_{2k} - \dot{\lambda}_{mn} \beta_{3m} \dot{\lambda}_{kn} \beta_{2k}, \quad \Omega_1 = \dot{\vec{\mathbf{e}}}_3^* \vec{\mathbf{e}}_2^* = -\dot{\lambda}_{mn} \beta_{3m} \lambda_{kn} \beta_{2k}, \\ \dot{\Omega}_2 &= \ddot{\lambda}_{mn} \beta_{3m} \lambda_{kn} \beta_{1k} + \dot{\lambda}_{mn} \beta_{3m} \dot{\lambda}_{kn} \beta_{1k}, \quad \Omega_2 = \dot{\vec{\mathbf{e}}}_3^* \vec{\mathbf{e}}_1^* = \dot{\lambda}_{mn} \beta_{3m} \lambda_{kn} \beta_{1k}, \\ \dot{\Omega}_3 &= \ddot{\lambda}_{mn} \beta_{1m} \lambda_{kn} \beta_{2k} + \dot{\lambda}_{mn} \beta_{1m} \dot{\lambda}_{kn} \beta_{2k}, \quad \Omega_3 = \dot{\vec{\mathbf{e}}}_1^* \vec{\mathbf{e}}_2^* = \dot{\lambda}_{mn} \beta_{1m} \lambda_{kn} \beta_{2k}, \end{aligned} \quad (6)$$

где точка обозначает производную по времени t .

Система уравнений движения тонкого упругого стержня, описывающая любые повороты поперечных сечений [8] при динамическом деформировании, имеет вид

$$U_{i,\xi} = (1 + \varepsilon) x_{j,\xi} \lambda_{ji} - x_{i,\xi}; \quad (7a)$$

$$\lambda_{ki,s} = (\lambda_{kr} \Delta \omega_{p,\xi} - \lambda_{kp} \Delta \omega_{r,\xi}), \Delta \omega_{j,\xi} = \lambda_{mj} \beta_{lm} C_{\ell\ell} \beta_{\ell k} \lambda_{kn} M_n, \\ \mathbf{C} = \begin{pmatrix} EJ_1^{-1} & 0 & 0 \\ 0 & EJ_2^{-1} & 0 \\ 0 & 0 & GJ_3^{-1} \end{pmatrix}; \quad (7б)$$

$$Q_{i,\xi} = [\rho F(\xi) \cdot (\ddot{U}_i + \ddot{R}_i) - q_i(\xi, \dots)] s_{,\xi} (1 + \varepsilon); \quad (7в)$$

$$M_{i,\xi} = \begin{pmatrix} \rho J_n(\xi) \dot{\Omega}_n \beta_{nk} \lambda_{ki} s_{,\xi} + \\ + \rho F(\xi) \cdot [(\ddot{U}_p + \ddot{R}_p) R_r - (\ddot{U}_r + \ddot{R}_r) R_p] s_{,\xi} - \\ - m_i(\xi, \dots) s_{,\xi} + x_{j,\xi} \lambda_{jr} Q_p - x_{j,\xi} \lambda_{jp} Q_r \end{pmatrix} (1 + \varepsilon), \begin{cases} i=1,2,3, \\ p=2,3,1, \\ r=3,1,2; \end{cases} \quad (7г)$$

$$\varepsilon = x_{j,\xi} \lambda_{jk} Q_k [EF(\xi) \cdot s_{,\xi}]^{-1} + \alpha \Delta T, \quad s_{,\xi} = (x_{k,\xi} x_{k,\xi})^{1/2},$$

где α – коэффициент температурного удлинения; ΔT – изменение температуры; нижний индекс ξ после запятой означает производную по этой переменной.

Система (7) состоит из 18 нелинейных дифференциальных уравнений первого порядка по ξ и второго порядка по времени t . Разрешающими функциями системы (7) являются 18 функций: $U_{1,2,3}(\xi, t)$ – проекции вектора перемещений точек осевой линии стержня на оси глобальной системы координат; $\lambda_{ij}(\xi, t)$ – компоненты матрицы, описывающей поворот главных осей инерции поперечных сечений при деформировании; $Q_{1,2,3}(\xi, t)$ – глобальные проекции вектора внутренних усилий; $M_{1,2,3}(\xi, t)$ – глобальные проекции внутренних моментов.

На девять компонентов матрицы поворота $\lambda(\xi, t)$ накладываются дополнительные зависимости, которые вытекают из общих свойств матриц поворота. Поэтому на практике вместо 9 компонент можно использовать только 3 функции в качестве параметров поворота, что

уменьшает число неизвестных функций и число уравнений с 18 до 12. В качестве параметров поворота могут выступать, например, углы Эйлера, самолетные углы, компоненты вектора конечного поворота [29, 30] и др. При использовании вектора конечного поворота угол поворота тройки ортов, связанных с поперечным сечением стержня, ограничивается величиной 2π [11] в общем случае. Для использования вектора конечного поворота в качестве параметра для любых поворотов при деформировании разработана специальная численная методика [8], суть которой будет представлена ниже. Если деформирование стержня укладывается в рамки ограничения поворота при деформировании 2π , тогда группа уравнений (7б) заменяется тремя уравнениями следующего вида:

$$\begin{aligned} \frac{\partial \lambda_{mn}}{\partial \omega_j} \frac{d\omega_j}{d\xi} \beta_{3m} \lambda_{kn} \beta_{2k} &= \frac{M_p \beta_{1k} \lambda_{kp}}{EJ_1} \frac{ds}{d\xi}, \\ \frac{\partial \lambda_{mn}}{\partial \omega_j} \frac{d\omega_j}{d\xi} \beta_{3m} \lambda_{kn} \beta_{1k} &= \frac{M_p \beta_{2k} \lambda_{kp}}{EJ_2} \frac{ds}{d\xi}, \\ \frac{\partial \lambda_{mn}}{\partial \omega_j} \frac{d\omega_j}{d\xi} \beta_{1m} \lambda_{kn} \beta_{2k} &= \frac{M_p \beta_{3k} \lambda_{kp}}{GJ_3} \frac{ds}{d\xi}, \end{aligned} \quad (8)$$

где $\vec{\omega} = \omega_i \vec{i}_i$ – вектор конечного поворота. Таким образом, имеем 12 дифференциальных уравнений и 12 неизвестных функций $U_{1,2,3}$, $\omega_{1,2,3}$, $Q_{1,2,3}$, $M_{1,2,3}$. Такое же количество уравнений и неизвестных получается при применении численной методики [8], но при этом ограничения вектора конечного поворота (при углах поворота кратных 2π определитель СЛАУ (8) равен нулю) не мешают получить численное решение краевой задачи.

Уравнения движения (7) являются системой нелинейных дифференциальных уравнений в частных производных. Для решения можно применять одношаговые и многошаговые методы прямого неявного интегрирования по времени [13, 14, 9] (методы Ньюмарка, Хоболта, Парка и др.). Основная идея численного интегрирования уравнений динамики стержня заключается в том, что производные по времени (скорости и ускорения) аппроксимируются специальными выражениями через координаты (перемещения и повороты). В эти выражения

также могут входить известные координаты, скорости и ускорения с предыдущих шагов по времени. Применение таких методов позволяет снизить размерность исходной задачи нелинейной динамики стержня, сводя ее, по существу, к задаче нелинейной статики.

Конфигурация стержня в каждый момент времени описывается следующими векторами-столбцами:

$$\mathbf{X}(\xi, t) = \{U_{1,2,3}(\xi, t), \omega_{1,2,3}(\xi, t), Q_{1,2,3}(\xi, t), M_{1,2,3}(\xi, t)\}^T, \dot{\mathbf{X}}(\xi, t), \ddot{\mathbf{X}}(\xi, t).$$

После проведения ряда тестов, авторы статьи пришли к выводу, что наиболее подходящим (по точности и устойчивости) для решения обозначенной задачи из рассмотренных методов прямого интегрирования является метод Парка [15, 13, 9]. Для расчетов используется модификация [9] этого метода:

$$\dot{\mathbf{X}}(t_1) = a_1 \mathbf{X}(t_{-3}) + a_2 \mathbf{X}(t_{-2}) + a_3 \mathbf{X}(t_{-1}) + a_4 \mathbf{X}(t_0) + a_5 \mathbf{X}(t_1), \quad (9)$$

где

$$a_1 = \frac{1}{4} \frac{\partial \dot{L}_5}{\partial f_1}, \quad a_2 = \frac{1}{4} \frac{\partial \dot{L}_4}{\partial f_1} + \frac{1}{4} \frac{\partial \dot{L}_5}{\partial f_2}, \quad a_3 = \frac{1}{2} \frac{\partial \dot{L}_3}{\partial f_1} + \frac{1}{4} \frac{\partial \dot{L}_4}{\partial f_2} + \frac{1}{4} \frac{\partial \dot{L}_5}{\partial f_3},$$

$$a_4 = \frac{1}{2} \frac{\partial \dot{L}_3}{\partial f_2} + \frac{1}{4} \frac{\partial \dot{L}_4}{\partial f_3} + \frac{1}{4} \frac{\partial \dot{L}_5}{\partial f_4}, \quad a_5 = \frac{1}{2} \frac{\partial \dot{L}_3}{\partial f_3} + \frac{1}{4} \frac{\partial \dot{L}_4}{\partial f_4} + \frac{1}{4} \frac{\partial \dot{L}_5}{\partial f_5}.$$

Здесь $L_n \equiv L_n(t_1)$, $L_n(x) = \sum_{j=1}^n \left(f_j \prod_{i=0, j \neq i}^n \frac{x - x_i}{x_j - x_i} \right)$ – полином Лагранжа для интерполяции по n точкам.

После замены скоростей и ускорений выражениями (9) в системе (7) остаются только первые производные по параметру ξ . Для замыкания задачи необходимо добавить краевые условия – 6 на левом конце и 6 на правом конце стержня.

2. Численная методика решения нелинейной краевой задачи

Представленная здесь численная методика решения нелинейной краевой задачи базируется на методике [18], которая в свое время породила несколько версий подпрограмм. Из числа известных и доступных подпрограмм это BVFPD [17] из библиотеки IMSL (International Mathematical and Statistical Library) и DD14AD из HSL Mathematical

Software Library. Помимо пользовательского интерфейса эти подпрограммы различаются алгоритмом работы в части критерия расходимости итераций метода Ньютона. Существенная разница в работе этих подпрограмм может быть устранена изменением текста DD14AD. Для поставленной задачи был разработан другой алгоритм, который соединил в себе основные идеи [18] и алгоритм продолжения решения по параметру [22, 23] (подпрограмма PASSIN), что оказалось незаменимым при реализации численной методики, позволяющей использовать вектор конечного поворота при любых деформациях осевой линии [8, 10]. Первая версия такого алгоритма представлена в работе [10], однако после сравнительного анализа процесса решения тестовых краевых задач подпрограммами KLPALG [10] и BVPDFD [18, 17] наш алгоритм претерпел существенные улучшения. Новая версия подпрограммы KLPALG, алгоритм которой представлен в этой статье, выполняет одинаковое с BVPDFD количество вызовов функций расчета невязок и факторизации матрицы Якоби на тестовых примерах, где возможно использование BVPDFD. На более сложных примерах, где BVPDFD выдает расхождение метода Ньютона, в подпрограмме KLPALG включается алгоритм продолжения решения по параметру [22, 23], который приводит к искомому решению.

Согласно [18] неизвестные разыскиваются в дискретных узлах, а дифференциальные уравнения аппроксимируются по правилу трапеций:

$$\frac{\mathbf{W}_{j+1} - \mathbf{W}_j}{h_j} = \frac{1}{2} [\mathbf{f}(\xi_j, \mathbf{W}_j) + \mathbf{f}(\xi_{j+1}, \mathbf{W}_{j+1})], \quad j=1, 2..N, \quad (10)$$

где в нашем случае системы уравнений стержня:

$0 = \xi_1 < \xi_2 < \dots < \xi_{N+1}$ – узлы, в которых разыскиваются значения неизвестных функций; $h_j = \xi_{j+1} - \xi_j$ – шаг сетки;

$$\mathbf{W}_j = \{U_1, U_2, U_3, \omega_1, \omega_2, \omega_3, Q_1, Q_2, Q_3, M_1, M_2, M_3\}^T \Big|_{\xi=\xi_j} \quad \text{– вектор-столбец 12 узловых неизвестных;}$$

– вектор-столбец 12 узловых неизвестных;

$\mathbf{f}(\xi, \mathbf{W})$ – правая часть системы дифференциальных уравнений движения тонкого упругого стержня (7)–(8) с учетом (9).

Краевые условия представляются в виде $\mathbf{g}_1(\mathbf{W}_1)=\mathbf{0}$, $\mathbf{g}_2(\mathbf{W}_1, \mathbf{W}_{N+1})=\mathbf{0}$, где $\mathbf{g}_{1,2}$ – векторы-столбцы, \mathbf{g}_1 имеет p строк, \mathbf{g}_2 имеет $(12-p)$ строк, q из которых связывают \mathbf{W}_1 и \mathbf{W}_{N+1} . Если начало стержня не связано с концом, тогда $p=6$, $q=0$, следовательно, \mathbf{g}_1 и \mathbf{g}_2 имеют по 6 строк (по 6 краевых условий на каждом конце стержня), таким образом, в дискретной форме краевая задача может быть представлена в следующем виде:

$$\mathbf{F}_\pi(\mathbf{W})=\mathbf{0}, \quad (11)$$

где

$$\mathbf{W} \equiv \left\{ \begin{array}{c} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \dots \\ \mathbf{W}_{N+1} \end{array} \right\} \equiv \left\{ \begin{array}{c} W_{1,1} \\ W_{2,1} \\ \dots \\ W_{12,1} \\ W_{1,2} \\ \dots \\ W_{12,N+1} \end{array} \right\} \equiv \left\{ \begin{array}{c} U_1^{(1)} \\ U_2^{(1)} \\ \dots \\ M_3^{(1)} \\ U_1^{(2)} \\ \dots \\ M_3^{(N+1)} \end{array} \right\},$$

$$\mathbf{F}_\pi(\mathbf{W}) \equiv \left\{ \begin{array}{c} \mathbf{g}_1(\mathbf{W}_1) \\ \mathbf{W}_2 - \mathbf{W}_1 - h_1(\mathbf{f}_1 + \mathbf{f}_2)/2 \\ \mathbf{W}_3 - \mathbf{W}_2 - h_2(\mathbf{f}_2 + \mathbf{f}_3)/2 \\ \dots \\ \mathbf{W}_{N+1} - \mathbf{W}_N - h_N(\mathbf{f}_N + \mathbf{f}_{N+1})/2 \\ \mathbf{g}_2(\mathbf{W}_1, \mathbf{W}_{N+1}) \end{array} \right\},$$

где $\mathbf{f}_j \equiv \mathbf{f}(\xi_j, \mathbf{W}_j)$, $h_j = \xi_{j+1} - \xi_j$, $j=1,2,\dots,N$.

Решение разыскивается итерационным способом:

$$\mathbf{W}^{v+1} = \mathbf{W}^v + \mu_v \cdot \Delta \mathbf{W}^v = \mathbf{W}^v + \mu_v \cdot \mathbf{F}_W(\mathbf{W}^v)^{-1} \mathbf{F}_\pi(\mathbf{W}^v), \quad (12)$$

где $\mathbf{F}_W(\mathbf{W}^v)$ – матрица Якоби системы уравнений (11) в точке \mathbf{W}^v для v -го приближения; μ_v – параметр контроля шага, определяется особым образом [18]. Матрица Якоби имеет следующую структуру:

$$\mathbf{F}_W(\mathbf{W}^v) = \begin{bmatrix} \mathbf{g}_{1W_1}(\mathbf{W}_1) & 0 & \dots & 0 \\ -\mathbf{I} - (\mathbf{f}_{1W}h_1)/2 & \mathbf{I} - (\mathbf{f}_{2W}h_1)/2 & \dots & 0 \\ 0 & -\mathbf{I} - (\mathbf{f}_{2W}h_2)/2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathbf{I} - (\mathbf{f}_{(N+1)W}h_N)/2 \\ \mathbf{g}_{2W_1}(\mathbf{W}_1, \mathbf{W}_{N+1}) & 0 & \dots & \mathbf{g}_{2W_{N+1}}(\mathbf{W}_1, \mathbf{W}_{N+1}) \end{bmatrix}.$$

Для определения неизвестных узловых значений функций используется система нелинейных уравнений (11). Численная методика [8] предполагает заменить в (11) уравнения для узловых значений проекций вектора конечного поворота

$$\omega_{k,j+1} - \omega_{k,j} - h_j (f_{3+k,j} + f_{3+k,j+1})/2 = 0, \quad k=1,2,3, \quad j=1,2,\dots,N, \quad (13)$$

на уравнения

$$\begin{aligned} & \left[\lambda_{1n,j+1} - \lambda_{1n,j} - h_j (f_{1n,j}^* + f_{1n,j+1}^*)/2 \right] (\lambda_{2n,j} + \lambda_{2n,j+1})/2 = 0, \\ & \left[\lambda_{1n,j+1} - \lambda_{1n,j} - h_j (f_{1n,j}^* + f_{1n,j+1}^*)/2 \right] (\lambda_{3n,j} + \lambda_{3n,j+1})/2 = 0, \quad (14) \\ & \left[\lambda_{2n,j+1} - \lambda_{2n,j} - h_j (f_{2n,j}^* + f_{2n,j+1}^*)/2 \right] (\lambda_{3n,j} + \lambda_{3n,j+1})/2 = 0, \end{aligned}$$

где по индексу n ведется суммирование от 1 до 3, а по индексу j нет суммирования. Этот индекс указывает на номер блока уравнений $1,2,\dots,N$.

В итерационном процессе (12) для поиска приращения $\Delta \mathbf{W}^v$ на каждом шаге решается система линейных алгебраических уравнений:

$$\mathbf{F}_W(\mathbf{W}^v) \Delta \mathbf{W}^v = -\mathbf{F}_\pi(\mathbf{W}^v). \quad (15)$$

Матрица Якоби $\mathbf{F}_W(\mathbf{W}^v)$ имеет следующую блочную трехдиагональную структуру:

$$\mathbf{F}_W(\mathbf{W}^v) \equiv \mathbf{A}^v \equiv \begin{pmatrix} \mathbf{A}_1 & \mathbf{C}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_2 & \mathbf{A}_2 & \mathbf{C}_2 & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_3 & \mathbf{A}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_N & \mathbf{C}_N \\ \mathbf{D} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_{N+1} & \mathbf{A}_{N+1} \end{pmatrix}, \quad (16)$$

где \mathbf{A}_k , \mathbf{B}_k , \mathbf{C}_k – матрицы 12×12 , нулями обозначены матрицы с нулевыми элементами, матрица $\mathbf{D} = \mathbf{g}_{2W_1}(\mathbf{W}_1, \mathbf{W}_{N+1})$ имеет q ненулевых элементов, если начало стержня связано с концом. Матрица \mathbf{B}_k имеет p ненулевых строк, \mathbf{C}_k имеет $(12 - p)$ ненулевых строк:

$$\mathbf{B}_k = \begin{pmatrix} \dots \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{C}_k = \begin{pmatrix} \mathbf{0} \\ \dots \end{pmatrix}.$$

Если $|\mathbf{A}^v| \neq 0$, тогда матрица (16) может быть представлена в виде разложения

$$\mathbf{A}^v \equiv \mathbf{L} \cdot \mathbf{U} \equiv \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \beta_2 & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \beta_3 & \mathbf{I} & \dots & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{0} \\ \delta_1 & \delta_2 & \delta_3 & \dots & (\beta_{N+1} + \delta_N) & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 & \mathbf{C}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \alpha_2 & \mathbf{C}_2 & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \alpha_3 & \dots & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \alpha_N & \mathbf{C}_N \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \alpha_{N+1} \end{pmatrix},$$

где матрицы α_k и β_k удовлетворяют следующим уравнениям:

$$\begin{aligned} \alpha_1 &= \mathbf{A}_1, \quad \alpha_j = \mathbf{A}_j - \beta_j \mathbf{C}_{j-1}, \quad \delta_1 \alpha_1 = \mathbf{D}, \quad j = 2, 3, \dots, N+1; \\ \beta_j \alpha_{j-1} &= \mathbf{B}_j, \quad \delta_j \alpha_j = -\delta_{j-1} \mathbf{C}_{j-1}, \quad j = 2, 3, \dots, N+1. \end{aligned} \quad (17)$$

Решение системы (15) сводится к решению двух систем

$$\mathbf{L} \cdot \mathbf{Y} = -\mathbf{F}_\pi(\mathbf{W}^v), \quad \mathbf{U} \cdot \Delta \mathbf{W}^v = \mathbf{Y}. \quad (18)$$

Алгоритм подпрограммы LU-факторизации матрицы Якоби следующий.

На входе в подпрограмму имеются массивы $\alpha_i = \mathbf{A}_i$, $\beta_i = \mathbf{B}_i$, \mathbf{C}_i , $\delta_i = \mathbf{D}$, они являются рабочими массивами и в них остается результат.

1. Начало цикла, $i=1$.

2. LU-факторизация матрицы α_i :

2.1. переставляются столбцы на первых p строках так, чтобы на главной диагонали стояли максимальные по модулю элементы;

2.2. переставляются строки оставшейся части матрицы α_i и матрицы β_i так, чтобы на главной диагонали стояли максимальные по модулю элементы (при этом строки могут меняться в матрицах \mathbf{C}_i и α_{i+1});

2.3. рассчитывается LU-разложение матрицы α_i , на главной диагонали которой стоят максимальные по модулю элементы.

3. Вычисляется $\delta_i = \delta_i \alpha_i^{-1}$, $\delta_{i+1} = -\delta_i \mathbf{C}_i$, $\beta_{i+1} = \beta_{i+1} \alpha_i^{-1}$, $\alpha_{i+1} = \alpha_{i+1} - \beta_{i+1} \mathbf{C}_i$.

4. $i=i+1$, если $i < N+1$ идем п. 2.

5. Вычисляется $\alpha_{N+1} = \alpha_{N+1} + \delta_{N+1}$.

6. Производится LU-разложение α_{N+1} , как в п. 2.3, при этом предварительно переставляются столбцы так, чтобы на главной диагонали стояли максимальные по модулю элементы.

На выходе из подпрограммы LU-факторизации матрицы Якоби имеем измененные массивы α_i , β_i , \mathbf{C}_i , δ_i и массивы перестановок столбцов и строк.

Для внесения в расчетную схему дополнительных сосредоточенных сил и моментов, приложенных не на концах стержня, необходимо добавить псевдоузел в точке приложения силы. То есть необходимо в вектор $\mathbf{F}_\pi(\mathbf{W})$ (11) внести дополнительное матричное уравнение, которое будет стыковать старый узел, в котором приложен сосредоточенный силовой фактор и новый псевдоузел, по расположению совпадающий со старым. Дополнительное матричное уравнение должно содержать равенство всех функций слева и справа, за исключением внутренних усилий или моментов, которые должны иметь скачок, равный приложенному внешнему сосредоточенному силовому фактору.

Увеличение числа дополнительно приложенных сил и моментов не изменит вид (16), просто возрастет количество неизвестных, а следовательно, в алгоритме решения полученной таким образом нелинейной системы ничего не изменится. Другими словами, процесс добавления сосредоточенных силовых факторов отразится на формировании нелинейных уравнений (11). По сути, такой способ эквивалентен разбиению стержня в местах приложения сил и моментов на несколько составных частей.

Стратегия выбора шага модифицированного метода Ньютона (12) следующая. Длина шага μ_v выбирается максимальной из ряда $\{1, 1/2, 1/4, \dots\}$, для которой выполняется условие сходимости

$$r(\mathbf{W}^v) - r(\mathbf{W}^v + \mu_v \Delta \mathbf{W}^v) \geq c \cdot \mu_v \cdot r(\mathbf{W}^v),$$

где $r(\mathbf{x}) = \|\mathbf{x}\|_2^2$, $\|\mathbf{X}\|_n = \sqrt{\sum (X_i)^n}$, $c = 1$ в [18] и $c = 0,5$ в [24, с. III.3-2] и в подпрограмме DD14AD. В новой версии подпрограммы KLPALG выбрана величина $c = 0,11$ при сравнении процессов расчета с BVFPD.

После того как решена система (11), полученное решение нужно скорректировать (уточнить) с учетом той ошибки, которую внесла аппроксимация производной по методу конечных разностей (10). Для этого используется метод отсроченной коррекции (deferred correction method) [18, 25, 26]. Рассмотрим кратко применение этого метода к нашей задаче (более подробные выкладки представлены в [10]).

Пусть \mathbf{W}^* – точное решение, тогда так называемая локальная ошибка аппроксимации (local truncation error) [18], обусловленная использованием приближенной формулы (10) с учетом $\mathbf{W}_j^{*l} \equiv \mathbf{f}(\xi_j, \mathbf{W}_j^*)$, имеет следующее выражение:

$$\tau_j = \sum_{k=0}^{\infty} a_k \frac{f^{(k)}(\mathbf{W}_{j+1/2}^*)}{k!} h_j^k, \quad j=1, 2, \dots, N, \quad \text{где } a_k = \frac{-k}{2^{(k+1)}(k+1)} [1 + (-1)^k]. \quad (19)$$

Необходимо отметить, что локальная ошибка аппроксимации (19) записана для точек, находящихся посередине между узловыми точками ξ_j , $j=1, 2, \dots, N+1$, шаг сетки в общем случае $h_j = \xi_{j+1} - \xi_j$, $j=1, 2, \dots, N$.

Получим такую аппроксимацию (19), которая предлагается в [25]. Пусть задана гладкая функция $y(\xi)$ и произвольная сетка с узлами ξ_j , $j=1,2,\dots,N+1$. Определим следующую функцию:

$$L(y, \xi) = \sum_{k=0}^m a_k \frac{y^{(k)}(\xi)}{k!} h(\xi)^k, \quad (20)$$

где $h(\xi)$ такая функция, что $h(\xi_{j+1/2}) \equiv h((\xi_{j+1} + \xi_j)/2) = \xi_{j+1} - \xi_j$, $j=1,2,\dots,N$.

Справедливо следующее равенство:

$$\tau(\xi) = L(W^*, \xi) + O(h^{m+1}).$$

Аппроксимируем (20) линейной комбинацией значений функции $y(\xi)$ в узловых точках:

$$M(y) = \sum_{i=1}^{m+1} w_i y(\xi_{i+r}), \quad (21)$$

где r – заданное целое значение (от которого зависит, какие узлы участвуют в аппроксимации); w_i – пока неизвестные весовые коэффициенты.

Пусть необходимо достигнуть точности порядка $O(h^{m+1})$. Разложим (21) в ряд Тейлора:

$$\begin{aligned} M(y) &= \sum_{i=1}^{m+1} w_i \sum_{j=0}^{\infty} \frac{y^{(j)}(\xi)}{j!} (\xi_{i+r} - \xi)^j = \\ &= \sum_{j=0}^m \left(\sum_{i=1}^{m+1} w_i \alpha_i^j \right) \frac{y^{(j)}(\xi)}{j!} h(\xi)^j + O(h(\xi)^{m+1}), \end{aligned} \quad (22)$$

где $\alpha_i = (\xi_{i+r} - \xi) / h(\xi)$.

Сравнивая (22) с (20), получаем СЛАУ относительно неизвестных весовых коэффициентов w_i :

$$\sum_{i=1}^{m+1} w_i \alpha_i^j = a_j, \quad j=0,1,\dots,m. \quad (23)$$

Находя w_i из (23), получим $M(y) = L(y, \xi) + O(h(\xi)^{m+1})$.

Система уравнений (23) является Вандермондовой системой уравнений [25]. Быстрый и эффективный алгоритм решения этой системы описан в [31], там же приведен текст подпрограммы на языке «Алгол». Таким образом, находя весовые коэффициенты из (23), можно по узловым значениям \mathbf{W} строить аппроксимацию локальной ошибки аппроксимации (19).

Метод отсроченной коррекции [25] предполагает приближенный расчет локальной ошибки аппроксимации (19) приближенно, по сути, с помощью метода конечных разностей (производные функции выражаются через узловые значения этой функции). Рассмотрим алгоритм расчета $\mathbf{S}_k(\mathbf{W})$ – конечно-разностной аппроксимации (19).

Разложение локальной ошибки аппроксимации (19) содержит члены, начиная со второй производной функции $\mathbf{f}(\mathbf{W}_{j+1/2}^*)$, и имеет только четные производные, следовательно, аппроксимация $\mathbf{S}_1(\mathbf{W})$ будет состояться по 4 узловым точкам, $\mathbf{S}_2(\mathbf{W})$ – по 6, $\mathbf{S}_3(\mathbf{W})$ – по 8, $\mathbf{S}_k(\mathbf{W})$ – по $(2k+2)$ узловым значениям. Поэтому корректное построение $\mathbf{S}_k(\mathbf{W})$ требует определенного числа разбиений по длине стержня.

Для нахождения $\mathbf{S}_k(\mathbf{W})$ в аппроксимации (21) следует положить $m = 2k + 1$ и

$$r(j) = \min[N - 2k - 1, \max(0, j - k - 1)], \quad j = 1, 2, \dots, N,$$

тогда коэффициенты α_i в (23) имеют вид

$$\alpha_i(j) = \left[\xi_{i+r(j)} - (\xi_j + \xi_{j+1}) / 2 \right] / (\xi_{j+1} - \xi_j). \quad (24)$$

Конечно-разностное представление $\mathbf{S}_k(\mathbf{W})$ локальной ошибки аппроксимации (19) для (14) имеет следующий вид:

$$\begin{aligned} \mathbf{S}_k(\omega_1) &= [\lambda_{2m,j} + \lambda_{2m,j+1}] \mathbf{S}_k(\lambda_{1m}) / 2, \\ \mathbf{S}_k(\omega_2) &= [\lambda_{3m,j} + \lambda_{3m,j+1}] \mathbf{S}_k(\lambda_{1m}) / 2, \end{aligned} \quad (25)$$

$$\mathbf{S}_k(\omega_3) = [\lambda_{3m,j} + \lambda_{3m,j+1}] \mathbf{S}_k(\lambda_{2m}) / 2.$$

Пусть \mathbf{W}^0 , рассчитанное с точностью $O(h^2)$ решение уравнения, а $\mathbf{S}_1(\mathbf{W}^0)$ – конечно-разностное представление локальной ошибки аппроксимации с точностью $O(h^2)$. Тогда после решения СЛАУ

$$\mathbf{F}_W(\mathbf{W}^0)\Delta = -\mathbf{S}_1(\mathbf{W}^0) \quad (26)$$

получаем аппроксимацию $O(h^2)$ глобальной ошибки $\mathbf{W}^* - \mathbf{W}^0$ (где \mathbf{W}^* – точное решение), т.е.

$$\Delta = \mathbf{W}^* - \mathbf{W}^0 + O(h^2).$$

Следует отметить, что для нахождения решения (26) не требуется много вычислений, так как после решения $\mathbf{F}_\pi(\mathbf{W}) = 0$ уже имеется LU-декомпозиция (17) матрицы Якоби $\mathbf{F}_W(\mathbf{W}^0)$.

Далее, решая нелинейную систему уравнений

$$\mathbf{F}_\pi(\mathbf{W}) = \mathbf{S}_1(\mathbf{W}^0), \quad (27)$$

получаем аппроксимацию решения с ошибкой $O(h^4)$, т.е. решение нелинейной проблемы \mathbf{W}^1 удовлетворяет следующему равенству:

$$\mathbf{W}^1 - \mathbf{W}^0 = O(h^4).$$

Это первый шаг метода отсроченной коррекции. Далее аналогично (26) для оценки текущей глобальной ошибки решаем СЛАУ:

$$\mathbf{F}_W(\mathbf{W}^1)\Delta = \mathbf{S}_1(\mathbf{W}^0) - \mathbf{S}_2(\mathbf{W}^1) \quad (28)$$

и получаем аппроксимацию $O(h^4)$ глобальной ошибки:

$$\Delta = \mathbf{W}^* - \mathbf{W}^2 + O(h^4).$$

Другой не менее важный аспект численной методики решения нелинейной краевой задачи – это алгоритм продолжения решения по параметру. В некоторых задачах, в частности при использовании в ка-

честве параметров поворота при деформировании на большие углы компоненты вектора конечного поворота [8, 10], сходимость метода Ньютона может быть очень медленной. Зачастую время расчета такого итерационного процесса при выборе длины ньютоновского шага (12) из условия сходимости метода стремится к бесконечности. Встает вопрос о выборе хорошего начального приближения. Подробно методика изложена в [23, 22, 10]. Здесь приведем основной алгоритм.

Процедура продолжения решения по параметру [10] имеет следующий вид.

1) Пусть дано решение $[\mathbf{W}_k, \gamma_k]$. Вычисляем $\frac{d\mathbf{W}_k}{d\gamma}$, решая СЛАУ

$$\mathbf{F}_W(\mathbf{W}_k, \gamma_k) \frac{d\mathbf{W}_k}{d\gamma} = -\mathbf{F}_\gamma(\mathbf{W}_k, \gamma_k) = -\mathbf{F}_\pi(\mathbf{W}_0). \quad (29)$$

2) Определяем знак Якобиана $sf_k = \text{sign}[\mathbf{F}_W(\mathbf{W}_k, \gamma_k)]$ (он меняется при переходе через точки бифуркации), вычисляем $\dot{\mathbf{W}}_k$ и $\dot{\gamma}_k$ по следующим формулам:

$$\dot{\gamma}_k = \frac{sf_0 \cdot sf_k}{\sqrt{1 + \theta \left(\frac{d\mathbf{W}_k^*}{d\gamma} \frac{d\mathbf{W}_k}{d\gamma} - 1 \right)}}, \quad \dot{\mathbf{W}}_k = \dot{\gamma}_k \frac{d\mathbf{W}_k}{d\gamma}, \quad (30)$$

где sf_0 – выбирается сначала при $\gamma = \gamma_0 = 0$ так, чтобы $\dot{\gamma}_0 > 0$, так как нам необходимо изменять параметр γ от 0 до 1; $\dot{W}^*(\sigma_0)$ такое, что $\dot{W}^*(\sigma_0)\dot{W}(\sigma_0) = \|\dot{W}(\sigma_0)\|^2$, $\theta \in (0, 1)$ – коэффициент.

3) Рассчитываем начальное приближение (делаем прогноз):

$$\begin{pmatrix} \mathbf{W}_{k+1}^0 \\ \gamma_{k+1}^0 \end{pmatrix} = \begin{pmatrix} \mathbf{W}_k \\ \gamma_k \end{pmatrix} + \begin{pmatrix} \dot{\mathbf{W}}_k \\ \dot{\gamma}_k \end{pmatrix} \Delta\sigma = \begin{pmatrix} \mathbf{W}_k \\ \gamma_k \end{pmatrix} + \begin{pmatrix} \Delta\mathbf{W}_k^0 \\ \Delta\gamma_k^0 \end{pmatrix}.$$

4) Для получения уточнения $\begin{pmatrix} \mathbf{W}_{k+1}^{v+1} \\ \gamma_{k+1}^{v+1} \end{pmatrix}$ (коррекции) имеем следующие выражения:

$$\mathbf{F}_W(\mathbf{W}_k, \gamma_k) \cdot z^v = -\mathbf{F}(\mathbf{W}_{k+1}^v, \gamma_{k+1}^v), \quad \mathbf{F}_W(\mathbf{W}_k, \gamma_k) \cdot y^v = -\mathbf{F}_\gamma(\mathbf{W}_{k+1}^v, \gamma_{k+1}^v),$$

$$\Delta\gamma^{v+1} = -\frac{N^v + \theta \cdot \dot{\mathbf{W}}_k^* \mathbf{z}^v}{(1-\theta)\dot{\gamma}_k - \theta \cdot \dot{\mathbf{W}}_k^* \mathbf{y}^v}, \quad \Delta\mathbf{W}^{v+1} = \mathbf{z}^v + \Delta\gamma^{v+1} \mathbf{y}^v,$$

$$\mathbf{W}_{k+1}^{v+1} = \mathbf{W}_k^v + \Delta\mathbf{W}^{v+1}, \quad \gamma_{k+1}^{v+1} = \gamma_k^v + \Delta\gamma^{v+1}. \quad (31)$$

Здесь

$$N^v \equiv N_3(\mathbf{W}(\sigma_1), \gamma(\sigma_1), \sigma_1) = \theta \cdot \dot{\mathbf{W}}^*(\sigma_0) [\mathbf{W}(\sigma_1) - \mathbf{W}(\sigma_0)] + \\ + (1-\theta) \cdot \dot{\gamma}(\sigma_0) [\gamma(\sigma_1) - \gamma(\sigma_0)] - (\sigma_1 - \sigma_0), \quad \sigma_1 = \sigma_0 + \Delta\sigma$$

или

$$N^v = \theta \cdot \dot{\mathbf{W}}_k^* [\mathbf{W}_{k+1}^v - \mathbf{W}_k] + (1-\theta) \cdot \dot{\gamma}_k [\gamma_{k+1}^v - \gamma_k] - \Delta\sigma.$$

Описанная процедура встраивается в основную процедуру решения нелинейной проблемы особым образом, алгоритм подпрограммы представлен ниже.

3. Алгоритм подпрограммы

На входе в подпрограмму имеется: начальное приближение решения W^0 ; N – число разбиений по длине стержня; ε – требуемая точность решения задачи; $\text{min}\mu$ – минимальный коэффициент ньютоновского шага ($0 < \text{min}\mu \leq 1$), ниже которого включается алгоритм продолжения решения по параметру; $\theta \in (0,1)$ – коэффициент для нормализации псевдодлины дуги (pseudoarclength normalization) [23, 22].

Инициализация рабочих переменных

$S_1 = S_1(W^0) \equiv 0$ – аппроксимация локальной ошибки усечения (local truncation error), вектор-столбец размерностью $12(N+1)$;

S_0 – вспомогательный рабочий вектор-столбец $12(N+1)$.

$k=0$ – счетчик алгоритма метода отсроченной коррекции (deferred corrections method);

$PM=0$ – алгоритм продолжения решения по параметру выключен;

$\gamma_0=0$, $\gamma_1=1$ – переменные для алгоритма продолжения решения по параметру;

$PE = \text{ложь}$ – метка стадии завершения работы алгоритма продолжения решения по параметру;

$br1 = \infty$ (некое большое число, например 10^{300}) – для контроля расхождения метода Ньютона;

$W = W^0$ – рабочий вектор-столбец размерностью $12(N+1)$;

$W2$ – вспомогательный рабочий вектор-столбец $12(N+1)$.

$ITNEW = 0$ – счетчик итераций метода Ньютона;

$NEPS = k \cdot h^2$, где k – маленькая константа, h – шаг сетки по переменной ξ ;

$bTOL = \infty$ – для контроля расхождения метода отсроченной коррекции;

$CASI = ложь$ – метка о выключении перерасчета Якобиана (квазиньютоновский метод).

Цикл итераций метода Ньютона

1. Если $PM > 0$, тогда $D = -F_{\pi}(W) - (\gamma_1 - 1)F_{\pi}(W^0)$, иначе $D = -F_{\pi}(W) + S_1$; $r = \|D\|_2^2$, $r1 = \sqrt{r}$;
2. Если $r1 < \varepsilon$, тогда {
 - 2.1. Если $PM = 0$, тогда идем в **25**;
 - 2.2. Если $PM > 1$, тогда { $PM = 3$; $CASI = ложь$ } ()
3. Если $CASI = правда$, тогда идем в **5**;
4. Расчет Якобиана $F_W(W)$, градиента $G(W)$, факторизация $F_W^{-1}(W)$;
5. Расчет $\Delta W = F_W^{-1}(W) \cdot D$;
6. Если $ERREV$, тогда идем в **18**;
7. Если $PM > 0$, тогда идем в пункт **33**;
8. Если $ITNEW > 0$, тогда $br1 = r1$;
9. $br2 = r1$;
10. $rG = -(G^T \cdot \Delta W)$;
11. Если $CASI = ложь$ и ($rG < 0$ или $|r - rG| > r$), тогда $\Delta W = -G$;
12. $W2 = W + \Delta W$, $\mu = 1$;
13. Расчет $F_{\pi}(W2)$, $D = -F_{\pi}(W2) + S_1$, $r = \|D\|_2^2$, $r1 = \sqrt{r}$;
14. Если $(br1 - r1) \leq 0,1 \mu \cdot br2$, тогда { (контроль шага)

14.1. $\mu = \mu \cdot 0.5$;

14.2. $W2 = W + \mu \cdot \Delta W$;

14.3. Если $\mu < \min \mu$, тогда {

Если $k > 0$, тогда {

Метод отсроченной коррекции разошелся, нужно уточнить сетку, возврат с ошибкой IERR=4}

Иначе {

Запускаем алгоритм продолжения решения по параметру

$PM = 1, \gamma_1 = 0$;

Идем в 1. } }

14.4. Идем в 13. }

15. $W = W2$;

16. Если $ITNEW < ITMAX$, тогда { $ITNEW = ITNEW + 1$, идем в пункт 2. }

17. Достигнут предел по количеству итераций, метод Ньютона разошелся или имеет слишком маленькую скорость сходимости, выход с ошибкой IERR=3.

Оценка точности решения краевой задачи

18. $ERREV = ложь$;

19. Расчет $cTOL = \|\Delta W\|_2^2$, $cTOL = \sqrt{cTOL}$;

20. Если $cTOL < TOL$, тогда { проблема решена успешно IERR=0, выход. }

21. Если $cTOL > bTOL$, тогда {
Метод отсроченной коррекции разошелся, нужно уточнить сетку, возврат с ошибкой IERR=4.}

22. $bTOL = cTOL$;

23. $ITNEW = ITNEW + 1$;

24. Идем в пункт 1.

Метод отсроченной коррекции (Deferred Correction Method)

25. Сохраняем $S_0 = S_1$;

26. $S_1 = S_{k+1}(W)$ по формулам (25);

27. $D = S_0 - S_1$;

28. $ERREV = истина$; (отправка на оценку достигнутой точности);

29. $k = k + 1$;
 30. $ITNEW = 0$, $br1 = 10^{300}$, $bTOL = 10^{300}$;
 31. $CASI = \text{истина}$;
 32. Идем в пункт 3.

Метод продолжения решения по параметру

33. Решаем СЛАНУ (29) $F_w(W) \frac{dW}{d\gamma} = -F_\pi(W^0)$;
 34. Знак Якобиана $sf = \text{sign}(F_w)$;

Корректор

35. Если $PM = 2$, тогда {

35.1. Вычисляем нормализацию N^v по формуле (31)

$$N^v = \theta \cdot \dot{W}_0^* [W - W2] + (1 - \theta) \cdot \dot{\gamma}_0 [\gamma_1 - \gamma_0] - \Delta\sigma ;$$

Вычисляем $\Delta\gamma_1$ по формулам (31):

$$\Delta\gamma_1 = -[N^v + \theta \cdot \dot{W}_0^* \Delta] / \left[(1 - \theta) \dot{\gamma}_0 - \theta \cdot \dot{W}_0^* \frac{dW}{d\gamma} \right] ,$$

$$\Delta = \Delta + \Delta\gamma_1 \frac{dW}{d\gamma} ; W = W + \Delta, \gamma_1 = \gamma_0 + \Delta\gamma_1 ;$$

$$r_2 = \|\Delta\|_\infty, \text{ где } \|X\|_n = \sqrt[n]{\sum (X_i)^n} ;$$

- 35.2. Если $r_2 > r_1$ или $\Delta\gamma_1 > \Delta\gamma_0$ или ($\gamma_1 < \gamma_0$ и $PE = \text{ложь}$),

тогда {

Отключаем стадию завершения алгоритма $PE = \text{ложь}$;

Уменьшаем шаг $\Delta\sigma = 0,5\Delta\sigma$;

$$\Delta\gamma_1 = \dot{\gamma}_0 \cdot \Delta\sigma, \gamma_1 = \gamma_0 + \Delta\gamma_1 ;$$

Если $\Delta\gamma_1 < \min\Delta\gamma$, тогда

{слишком маленький шаг по параметру продолжения ,
 $IERR=3$, выход.}

$$W = W2 + \dot{W}_0 \Delta\sigma ;$$

Идем в 1.}

- 35.3. $r_1 = r_2$;

- 35.4. Идем в пункт 1.}

Предиктор

36. Если $PM = 3$, тогда {

36.1. Вычисляем \dot{W}_0 и $\dot{\gamma}_0$ по формулам (30)

$$\dot{\gamma}_0 = sf_0 \cdot sf \cdot \left[1 + \theta \left(\frac{dW^*}{d\gamma} \frac{dW}{d\gamma} - 1 \right) \right]^{-1/2}, \quad \dot{W}_0 = \dot{\gamma}_0 \frac{dW}{d\gamma}, \quad \Delta\gamma_0 = \dot{\gamma}_0 \cdot \Delta\sigma;$$

36.2. Если $\Delta\gamma_0 < 0$, тогда $sf_0 = -sf_0$, идем в пункт 36.1;

36.3. $\gamma_2 = \gamma_1 + \Delta\gamma_0$;

36.3. Если ($PE = ложь$ и $\gamma_2 > 1$) или $PE = истина$, тогда {

Ставим алгоритм на завершение $PE = истина$

Целим в единицу: $\Delta\sigma = (1 - \gamma_1) / \dot{\gamma}_0$;

$\Delta\gamma_0 = \dot{\gamma}_0 \cdot \Delta\sigma$, $\gamma_2 = \gamma_1 + \Delta\gamma_0$;

Если $|\Delta\gamma_0| < \delta$, тогда $\gamma_1 = 1$, $PM = 0$, идем в пункт 25. }

36.4. $\gamma_0 = \gamma_1$, $\gamma_1 = \gamma_2$;

36.5. Запоминаем текущий вектор $W_2 = W$;

36.6. Делаем прогноз: $W = W + \dot{W}_0 \Delta\sigma$;

36.7. Включаем алгоритм на коррекцию $PM = 2$;

36.8. $r_1 = \infty$ (некое большое число, например 10^{300});

36.9. Идем в пункт 1; }

Предиктор, первый шаг

37. Если $PM = 1$, тогда {

37.1. Вычисляем \dot{W}_0 и $\dot{\gamma}_0$ по формулам (30)

$$\dot{\gamma}_0 = \left[1 + \theta \left(\frac{dW^*}{d\gamma} \frac{dW}{d\gamma} - 1 \right) \right]^{-1/2}, \quad \dot{W}_0 = \dot{\gamma}_0 \frac{dW}{d\gamma};$$

37.2. Выбираем самый первый шаг $\Delta\sigma = 1.0 / \dot{\gamma}_0$ (условно из расчета, что потребуется один шаг для достижения $\gamma = 1$);

37.3. $\Delta\gamma_0 = \dot{\gamma}_0 \cdot \Delta\sigma$;

37.4. Если $\Delta\gamma_0 < 0$, тогда $sf_0 = -1$, $\Delta\gamma_0 = -\Delta\gamma_0$; иначе $sf_0 = 1$;

37.5. Запоминаем текущий вектор $W_2 = W$;

37.5. Делаем прогноз: $W = W + \dot{W}_0 \Delta\sigma$, $\gamma_1 = \gamma_0 + \Delta\gamma_0$;

37.6. Ставим на коррекцию алгоритм продолжения решения
 $PM = 2$;

37.7. $r_1 = \infty$ (некое большое число, например 10^{300});

37.8. Идем в пункт 1. }

Конец алгоритма.

Основное отличие DD14AD от BVFPD – это наличие в последней условия как в пункте 8 приведенного выше алгоритма. Это условие, по сути, отменяет проверку сходимости после первого шага метода Ньютона.

Выводы

Тестирование данного алгоритма проводилось на достаточно большом количестве задач. Например, были повторены расчеты, опубликованные в работах [28, 32]. Точных тестов на скорость не проводилось, но предварительно можно заключить, что разработанная новая подпрограмма KLPALG, алгоритм которой представлен в данной статье, не уступает DD14AD и BVFPD по скорости исполнения кода и способна решать более широкий круг нелинейных краевых задач. Новизна представленных в данной статье результатов заключается: во-первых, в обобщении результатов многолетней работы авторов данной статьи в области механики тонких криволинейных стержней (представлена краткая сводка основных формул с описанием); во-вторых, представлен новый алгоритм подпрограммы KLPALG и описана используемая численная методика, что может помочь читателю в сжатые сроки ознакомиться, понять и реализовать в виде подпрограммы большой массив информации, представленный в прошлом столетии в ряде публикаций на английском языке, в том числе с ошибками, опечатками и недомолвками.

В представленной методике не рассмотрен вопрос о выборе оптимальной сетки, этот алгоритм [18] планируется внедрить в KLPALG. Другой нерассмотренный аспект подпрограмм BVFPD и DD14AD – это процедура продолжения решения, когда решение предыдущей краевой задачи экстраполируется вперед по методу Эйлера и при этом происходит программный выбор длины такого шага [33, 34]. Такой подход оказался неработоспособен в задачах [8, 10], поэтому в подпрограмму KLPALG был внедрен другой алгоритм продолжения решения по параметру [23, 22, 10].

Работа выполнена при финансовой поддержке Министерства образования и науки РФ по государственному заданию №2014/138, проект № 435.

Библиографический список

1. Светлицкий В.А. Механика стержней: учебник для вузов: в 2 ч. Ч. 1. Статика. – М.: Высш. шк., 1987. – 320 с.
2. Жилин П.А. Прикладная механика. Теория тонких упругих стержней: учеб. пособие. – СПб.: Изд-во Политехн. ун-та, 2007. – 101 с.
3. Светлицкий В.А. Строительная механика машин. Механика стержней: в 2 т. Т. 2. Динамика. – М.: Физматлит, 2009. – 383 с.
4. Kirchhoff G.R. Ueber das Gleichgewicht und die Bewegung einer elastischen Staben // Crelle Journal fuer die reine und angewandte Mathematik. – 1858. – Bd. 56. – S. 285–313.
5. Кирхгоф Г. Механика. – М.: Изд-во АН СССР, 1962. – 402 с.
6. Ляв А. Математическая теория упругости.– М.; Л.: Объединение научно-технических издательств, 1935. – 674 с.
7. Пустовой Н.В., Левин В.Е., Красноруцкий Д.А. Применение геометрически нелинейных уравнений стержня к расчету статики и динамики тросов. Ч. 1 // Научный вестник Новосиб. гос. техн. ун-та, 2012. – № 1 (46). – С. 83–92.
8. Пустовой Н.В., Левин В.Е., Красноруцкий Д.А. Методика вычисления параметров больших поворотов поперечных сечений гибкого стержня при расчетах в рамках его дифференциальной модели. Ч. 1 // Научный вестник Новосиб. гос. техн. ун-та. – 2013. – № 2 (51). – С. 155–164.
9. Красноруцкий Д.А., Левин В.Е., Пустовой Н.В. Нелинейная динамика тонких упругих стержней // Нелинейные колебания механических систем: труды IX Всерос. науч. конф. (Нижний Новгород, 24–29 сентября 2012 г.) / под ред. Д.В. Баландина, В.И. Ерофеева, И.С. Павлова. – Н. Новгород: Наш дом, 2012. – С. 557–565.
10. Пустовой Н.В., Левин В.Е., Красноруцкий Д.А. Методика вычисления параметров больших поворотов поперечных сечений гибкого стержня при расчетах в рамках его дифференциальной модели. Ч. 2 // Научный вестник Новосиб. гос. техн. ун-та. – 2013. – № 3(52). – С. 146–159.

11. Сорокин Ф.Д. Прямое тензорное представление уравнений больших перемещений гибкого стержня с использованием вектора конечного поворота // Известия РАН. МТТ. – 1994. – № 1. – С. 164–168.

12. Nonlinear dynamic deformation simulation for helical rod like objects / H. Du, W. Xiong, H. Wang, Z. Wang, B. Yuan // Engineering Review. – 2013. – Vol. 33. – Iss. 3. – P. 233–238.

13. Bathe K.J. Finite Element Procedures. Englewood Cliffs. – NY: Prentice Hall, 1996. – 1037 p.

14. Newmark N.M. A Method of Computation for Structural Dynamics // Journal of Engineering Mechanics Division, ASCE. – 1959. – Vol. 85. – No. EM3. – P. 67–94.

15. Park K.C. An improved stiffly stable method for direct integration of nonlinear structural dynamic equations // Journal of Applied Mechanics, ASME. – 1975. – Vol.42. – Iss. 2. – P. 464–470.

16. Shampine L.F., Muir P.H., Xu H. A User-Friendly Fortran BVP Solver // Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM). – 2006. – Vol. 1. – No. 2. – P. 201–217.

17. IMSL: Fortran Numerical Library. User's Guide. Math Library. Version. 7.0, available at: http://www.roguewave.com/documents.aspx?entryid=519&comma nd=core_download. – Date 27.05.2014. – Title from screen.

18. Pereyra V. Pasva3: An adaptive finite difference fortran program for first order nonlinear, ordinary boundary problems // Lecture Notes in Computer Science. – 1979. – Vol. 76. – P. 67–88.

19. Rashidinia J. Finite difference methods for a class of two-point boundary value problems // IUST International Journal of Engineering Science. – 2008. – Vol. 19. – No. 5–2. – P. 67–72.

20. Вайнберг А.М. Математическое моделирование процессов переноса. Решение нелинейных краевых задач. – М.; Иерусалим, 2009. – 209 с.

21. Dinkar Sharma, Ram Jiwari, Sheo Kumar. Numerical Solution of Two Point Boundary Value Problems Using Galerkin-Finite Element Method // International Journal of Nonlinear Science. – 2012. – Vol. 13. – No. 2. – P. 204–210.

22. Lentini M. Boundary Value Problems over Semi-Infinite Intervals: Ph.D. Thesis / Cal. Inst, of Technology. – 1978. – 123 p.

23. Keller H.B. Constructive Methods for Bifurcation and Nonlinear Eigenvalue Problems // *Lecture Notes in Mathematics*, 704. – Springer-Verlag Berlin Heidelberg, New York, 1979. – P. 241–251.

24. Pereyra V., Keller H.B. Finite Difference Solution of Two-Point Boundary Value Problems: Preprint 69 / Dept. Math., Univ. – Southern California. – Los Angeles, 1976. – 130 p.

25. Pereyra V. High Order Finite Difference Solution of Differential Equations. – Stanford Univ. Comp. Sci. Report STAN-CS-73-348, 1973. – 86 p.

26. Lentini M., Pereyra V. An adaptive finite difference solver for nonlinear two point boundary problems with mild boundary layers // *SIAM J. Numer. Anal.* – 1977. – Vol. 14. – No. 1. – P. 91–111.

27. Красноруцкий Д.А. Развитие модели тонкого упругого стержня для расчета изгибно-крутильных колебаний авиационных лопастей // *Наука. Промышленность. Оборона: тр. 13 Всерос. науч.-техн. конф.; Новосиб. гос. техн. ун-т.* – Новосибирск, 2012. – С. 328–332.

28. Пустовой Н.В., Левин В.Е., Красноруцкий Д.А. Математическое моделирование контактного взаимодействия витков гибкого стержня при петлеобразовании // *Прикладные задачи математики: материалы 21-й Междунар. науч.-техн. конф. (Севастополь, 16–20 сент. 2013 г.).* – Севастополь: Изд-во Севастоп. нац. техн. ун-та, 2013. – С. 47–51.

29. Аргирис Дж. Современные достижения в методах расчета конструкций с применением матриц. – М.: Изд-во лит. по строительству, 1968. – 242 с.

30. Argyris J.H. An excursion into large rotations // *Comp. Meth. Appl. Mech. Eng.* – 1982. – Vol. 32. – No. 1. – P. 85–155.

31. Björck A., Pereyra V. Solution of Vandermonde Systems of Equations // *Mathematics of computation.* – 1970. – Vol. 24. – No. 112. – P. 893–903.

32. Пустовой Н.В., Левин В.Е., Красноруцкий Д.А. Применение геометрически нелинейных уравнений стержня к расчету статики и динамики тросов. Ч. 2 // *Научный вестник Новосиб. гос. техн. ун-та.* – 2012. – № 2. – С. 106–116.

33. Ortega J.M. and Rheinboldt W.C. Iterative Solution of Nonlinear Equations in Several Variables. – New York: Academic Press, 1970. – 572 p.

34. Deuffhard P. A Stepsize Control for Continuation Methods and its Special Application to Multiple Shooting Techniques // *Mathematik.* – 1979. – P. 115–146.

References

1. Svetlitskii V.A. *Mekhanika sterzhnei* [Mechanics of rods]. Statika. Moscow: Vysshaia shkola, 1987. 320 p.

2. Zhilin P.A. *Prikladnaia mekhanika. Teoriia tonkikh uprugikh sterzhnei: uchebnoe posobie* [Applied mechanics. The theory of thin elastic rods]. Sankt-Peterburgskiy politekhnicheskii universitet, 2007. 101 p.

3. Svetlitskii V.A. *Stroitel'naya mekhanika mashin. Mekhanika sterzhnei* [Construction mechanics of machines. Mechanics of rods]. Dinamika. Moscow: Fizmatlit, 2009. 383 p.

4. Kirchhoff G.R. Ueber das Gleichgewicht und die Bewegung einer elastischen Staben [About the equilibrium and motion of an elastic rods]. *Crelle Journal fuer die reine und angewandte Mathematik*, 1858, Bd. 56, pp. 285-313.

5. Kirkhgof G. *Mekhanika* [Mechanics]. Akademiia nauk SSSR. Moscow, 1962. 402 p.

6. Liav A. *Matematicheskaiia teoriia uprugosti* [A treatise on the mathematical theory of elasticity]. Moscow; Leningrad: Otdelenie nauchno-tekhnicheskikh izdatelstv, 1935. 674 p.

7. Pustovoy N.V., Levin V.E., Krasnorutskiy D.A. *Primenenie geometricheski nelineinykh uravnenii sterzhnia k raschetu statiki i dinamiki trosov. Chast 1* [Applying of geometrically nonlinear equations of a rod to calculate statics and dynamics of cables. Part 1]. *NSTU Scientific Bulletin*, 2012, no. 1 (46), pp. 83-92.

8. Pustovoy N.V., Levin V.E., Krasnorutskiy D.A. *Metodika vychisleniia parametrov bol'shikh povorotov poperechnykh sechenii gibkogo sterzhnia pri raschetakh v ramkakh ego differentsial'noi modeli. Chast 1* [The method of calculation of parameters of big rotations of cross-sections of a flexible rod using its differential model. Part 1]. *NSTU Scientific Bulletin*, 2013, no. 2 (51), pp. 155-164.

9. Krasnorutskiy D.A., Levin V.E., Pustovoy N.V. *Nelineinaia dinamika tonkikh uprugikh sterzhnei* [Nonlinear dynamics of thin elastic rods]. *Nelineinye kolebaniya mekhanicheskikh sistem: trudy IX Vserossiyskoy nauchnoi konferentsii (Nizhniy Novgorod, 24–29 sentyabrya 2012 g.)*. N. Novgorod: Nash dom, 2012, pp. 557-565.

10. Pustovoy N.V., Levin V.E., Krasnorutskiy D.A. Metodika vychisleniia parametrov bolshikh povorotov poperechnykh sechenii gibkogo sterzhnia pri raschetakh v ramkakh ego differentsial'noi modeli. Chast 2. [The method of calculation of parameters of big rotations of cross-sections of a flexible rod using its differential model. Part 2]. *NSTU Scientific Bulletin*, 2013, no. 3(52), pp. 146-159.

11. Sorokin F.D. Priamoe tenzornoe predstavlenie uravnenii bol'shikh peremeshchenii gibkogo sterzhnia s ispol'zovaniem vektora konechnogo povorota [Direct tensor representation of the equations large displacements of an elastic rod with using the vector of finite rotation]. *MTT*, 1994, no. 1, pp. 164-168.

12. Du H., Xiong W., Wang H., Wang Z., Yuan B. Nonlinear dynamic deformation simulation for helical rod like objects. *Engineering Review*, 2013, vol. 33, iss. 3, pp. 233-238.

13. Bathe K.J. Finite Element Procedures. Englewood Cliffs. NY: Prentice Hall, 1996. 1037 p.

14. Newmark N.M. A Method of Computation for Structural Dynamics. *Journal of Engineering Mechanics Division, ASCE*, July 1959, vol. 85, no. EM3, pp. 67-94.

15. Park K.S. An improved stiffly stable method for direct integration of nonlinear structural dynamic equations. *Journal of Applied Mechanics, ASME*, June 1975, vol. 42, iss. 2, pp. 464-470.

16. Shampine L.F., Muir P.H., Xu H. A User-Friendly Fortran BVP Solver. *Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM)*, 2006, vol. 1, no. 2, pp. 201-217.

17. IMSL [electronic resource]: Fortran Numerical Library. User's Guide. Math Library. Version. 7.0, available at: http://www.rogue-wave.com/documents.aspx?entryid=519&command=core_download.

18. Pereyra V. Pasva3: An adaptive finite difference fortran program for first order nonlinear, ordinary boundary problems. *Lecture Notes in Computer Science*, 1979, vol. 76, pp 67-88.

19. Rashidinia J. Finite difference methods for a class of two-point boundary value problems. *IUST International Journal of Engineering Science*, 2008, vol. 19, no. 5-2, pp. 67-72.

20. Vainberg A.M. Matematicheskoe modelirovanie protsessov perenosnosa. Reshenie nelineinykh kraevykh zadach [Computer-aided simulation of transfer processes. Solving of a nonlinear boundary-value problems]. Moskow-Jerusalem, 2009. 209 p.

21. Dinkar Sharma, Ram Jiwari, Sheo Kumar. Numerical Solution of Two Point Boundary Value Problems Using Galerkin-Finite Element Method. *International Journal of Nonlinear Science*, 2012, vol. 13, no. 2, pp. 204-210.

22. Lentini M. Boundary Value Problems over Semi-Infinite Intervals: Ph.D. Thesis, Cal. Inst. of Technology, 1978, 123 p.

23. Keller H.B. Constructive Methods for Bifurcation and Nonlinear Eigenvalue Problems. *Lecture Notes in Mathematics*, 704. Springer-Verlag Berlin Heidelberg, New York, 1979, pp. 241-251.

24. Pereyra V., Keller H.B. Finite Difference Solution of Two-Point Boundary Value Problems: Preprint 69. Dept. Math., Univ. Southern California, Los Angeles, 1976, 130 p.

25. Pereyra V. High Order Finite Difference Solution of Differential Equations, Stanford Univ. Comp. Sci. Report STAN-CS-73-348, 1973, 86 p.

26. Lentini M., Pereyra V. An adaptive finite difference solver for nonlinear two point boundary problems with mild boundary layers. *SIAM J. Numer. Anal.*, 1977, vol. 14, no. 1, pp. 91-111.

27. Krasnorutskiy D.A. Razvitie modeli tonkogo uprugogo sterzhnia dlia rascheta izgibno-krutil'nykh kolebaniy aviatsionnykh lopastei [Development of a thin elastic rod model for the calculation of flexural-torsional vibration of aircraft blades]. *Trudy 13 Vserossiiskoi nauchno-tekhnicheskoj konferentsii "Nauka. Promyshlennost. Oborona"*. Novosibirskiy gosudarstvennyi tekhnicheskij universitet, 2012, pp. 328-332.

28. Pustovoy N.V., Levin V.E., Krasnorutskiy D.A. Matematicheskoe modelirovanie kontaktnogo vzaimodeistviia vitkov gibkogo sterzhnia pri petleobrazovanii. [Mathematical modeling of self-contact at a looping process of flexible rod]. *Materialy 21 mezhdunarodnoi nauchno-tekhnicheskoj konferentsii "Prikladnye zadachi matematiki" (Sevastopol', 16–20 Sent. 2013 g.)*. Sevastopolskiy natsionalnyi tekhnicheskij universitet, 2013, pp. 47-51.

29. Argiris Dzh. Sovremennye dostizheniia v metodakh rascheta konstruksii s primeneniem matrits [Recent developments in the methods of calculation of structures using matrixes]. Moscow: Izdatelstvo literatury po stroitelstvu, 1968. 242 p.

30. Argyris J.H. An excursion into large rotations. *Comp. Meth. Appl. Mech. Eng.*, 1982, vol. 32, no. 1, pp. 85-155.

31. Björck A., Pereyra V. Solution of Vandermonde Systems of Equations. *Mathematics of computation*, 1970, vol. 24, no. 112, pp. 893-903

32. Pustovoy N.V., Levin V.E., Krasnorutskiy D.A. Primenenie geometricheski nelineinykh uravnenii sterzhnia k raschetu statiki i dinamiki trosov. Chast 2 [Applying of geometrically nonlinear equations of a rod to calculate statics and dynamics of cables. Part 2]. *NSTU Scientific Bulletin*, 2012, no. 2, pp. 106-116.

33. Ortega J.M. and Rheinboldt W.C. Iterative Solution of Nonlinear Equations in Several Variables. New York: Academic Press, 1970. 572 p.

34. Deuflhard P. A Stepsize Control for Continuation Methods and its Special Application to Multiple Shooting Techniques. *Mathematik*, 1979, pp. 115-146

Об авторах

Пустовой Николай Васильевич (Новосибирск, Россия) – доктор технических наук, профессор, заведующий кафедрой «Прочность летательных аппаратов», ректор Новосибирского государственного технического университета (630073, г. Новосибирск, пр. К. Маркса, 20, e-mail: rector@nstu.ru).

Левин Владимир Евгеньевич (Новосибирск, Россия) – доктор технических наук, профессор, заместитель заведующего кафедрой «Прочность летательных аппаратов» Новосибирского государственного технического университета (630073, г. Новосибирск, пр. К. Маркса, 20, e-mail: levin@craft.nstu.ru).

Красноруцкий Дмитрий Александрович (Новосибирск, Россия) – кандидат технических наук, доцент кафедры «Прочность летательных аппаратов» Новосибирского государственного технического университета (630073, г. Новосибирск, пр. К. Маркса, 20, e-mail: krasnorutskiy@corp.nstu.ru).

About the authors

Nikolay V. Pustovoy (Novosibirsk, Russian Federation) – Doctor of Technical Sciences, Professor, Head of the Department of Aircraft Strength, rector of Novosibirsk State Technical University (20, K. Marks av., 630073, Novosibirsk, Russian Federation, e-mail: rector@nstu.ru).

Vladimir E. Levin (Novosibirsk, Russian Federation) – Doctor of Technical Sciences, Professor, Deputy head of the Department of Aircraft Strength of Novosibirsk State Technical University (20, K. Marks av., 630073, Novosibirsk, Russian Federation, e-mail: levin@craft.nstu.ru).

Dmitry A. Krasnorutskiy (Novosibirsk, Russian Federation) – Ph. D. in Technical Sciences, Associate Professor of the Department of Aircraft Strength of Novosibirsk State Technical University (20, K. Marks av., 630073, Novosibirsk, Russian Federation, e-mail: krasnorutskiy@corp.nstu.ru).

Получено 04.04.2014

Просьба ссылаться на эту статью в русскоязычных источниках следующим образом:

Пустовой Н.В., Левин В.Е., Красноруцкий Д.А. Алгоритм численного решения нелинейной краевой задачи динамического деформирования тонкого стержня // Вестник Пермского национального исследовательского политехнического университета. Механика. – 2014. – № 2. – С. 168–199.

Please cite this article in English as:

Pustovoy N.V., Levin V.E., Krasnorutskiy D.A. The numerical algorithm for solving nonlinear boundary problem of thin rod's dynamic deformations. *PNRPU Mechanics Bulletin*. 2014. No. 2. P. 168-199.