

Selection of a Method for Solving Nonlinear Equations in Shallow-Water Icing Model Implementation

A. D. Bagrov^{1,*} and A. A. Rybakov^{1,**}

(Submitted by TODO)

¹ Joint Supercomputer Center of the Russian Academy of Sciences – branch of Scientific Research Institute of System Analysis of the Russian Academy of Sciences, Leninsky prospect 32a, Moscow, 119334, Russia

Received TODO

Abstract—Моделирование ледяных наростов на профилях летательных аппаратов в процессе их полета в среде, содержащей переохлажденные капли воды, является крайней важной задачей для безопасности полетов, так как образуемые ледяные наросты существенным образом влияют на летные характеристики. В одной из моделей решения поставленной задачи – shallow-water icing model (SWIM) – центральную роль в численном моделировании занимает задача решения нелинейных уравнений с одной переменной. Так как данная задача занимает подавляюще большую часть всех расчетов, то вопрос выбора оптимального метода решения нелинейных уравнений и оптимизации данных методов встает особенно остро. В данной статье описан анализ использования различных методов решения нелинейных уравнений в реализации решателя SWIM с учетом особенностей решаемых уравнений и продемонстрировано существенное ускорение расчетных кодов при выполнении вычислений на суперкомпьютерах МСЦ РАН.

2010 Mathematical Subject Classification: 65H05, 65Y20

Keywords and phrases: *nonlinear equations, shallow-water icing model, метод бисекций, метод ньютона, метод брента, TODO*

1. INTRODUCTION

В настоящее время известно множество расчетных кодов, использующихся для численного моделирования обледенения поверхности обтекаемого тела. Одними из наиболее популярных пакетов для решения данной задачи являются Lewice [1] и ONERA. В данной статье мы не будем касаться особенностей данных расчетных пакетов и различий между ними, а рассмотрим только реализацию решателя SWIM, подробно описанного в [2]. При выполнении компьютерного моделировании процесса обледенения поверхности, обтекаемой свободным потоком, в shallow-water icing model выполняется одновременный расчет нарастания льда и течения пленки жидкости по поверхности обтекаемого тела. При этом учитывается выпадение влаги на поверхность тела, испарение влаги или сублимация льда с поверхности тела, течение жидкой пленки по поверхности с частичным ее замерзанием, а также перетекание потоков тепла между телом и поверхностью и между поверхностью и окружающим воздухом.

Численные расчеты выполняются на поверхностной расчетной сетке, состоящей из отдельных ячеек, при этом в каждой ячейке поверхности должен выполняться закон сохранения массы, записываемый в следующем виде:

$$\rho_w \left[\frac{\partial h_f}{\partial t} + \text{div}(\bar{u}h_f) \right] = U_\infty LWC\beta - \dot{m}_{evap} - \dot{m}_{ice} \quad (1)$$

* E-mail: andrey.bagrov@yandex.ru

** E-mail: rybakov.aax@gmail.com

В данной формуле ρ_w – плотность воды, h_f – высота водяной пленки на поверхности, \bar{u} – скорость течения водяной пленки, U_∞ – скорость набегающего свободного потока, LWC – содержание влаги в набегающем потоке, \dot{m}_{evap} – удельная скорость испарения или сублимации с поверхности обтекаемого тела, \dot{m}_{ice} – удельная скорость нарастания массы льда.

Кроме того, в каждой ячейке выполняется закон сохранения энергии:

$$\rho_w \left[\frac{\partial h_f C_w \tilde{T}}{\partial t} + \text{div}(\bar{u} h_f C_w \tilde{T}) \right] = \left[C_w \tilde{T}_{d,\infty} + \frac{\|\bar{u}_d\|^2}{2} \right] \times U_\infty LWC \beta$$

$$- \frac{1}{2} (L_{evap} + L_{subl}) \dot{m}_{evap} + (L_{fusion} - C_{ice} \tilde{T}) \dot{m}_{ice} + \sigma (T_\infty^4 - T^4) + \dot{Q}_h + \dot{Q}_{cond}$$
(2)

В данной формуле C_w – удельная теплоемкость воды, \tilde{T} – температура ячейки в градусах Цельсия, $\tilde{T}_{d,\infty}$ – температура выпадающей влаги в градусах Цельсия, \bar{u}_d – скорость выпадающих на поверхность капель, L_{evap} – скрытая теплота испарения воды, L_{subl} – скрытая теплота сублимации льда, L_{fusion} – скрытая теплота плавления льда, C_{ice} – теплоемкость льда, σ – постоянная Больцмана, T_∞ – температура свободного потока в кельвинах, T – температура ячейки в кельвинах, \dot{Q}_h – удельная величина потока тепла, получаемого из воздуха, \dot{Q}_{cond} – удельная величина потока тепла, поступающего в ячейку из обтекаемого тела.

Для численного решения приведенной системы уравнений необходимо выполнить ее дискретизацию по времени и пространству, как это показано в [3]. После этого получим систему из двух разностных уравнений, в которые входят три неизвестные величины: температура поверхности \tilde{T} , высота водяной пленки h_f и высота ледяного нароста h_{ice} .

Также при решении системы уравнений должны выполняться условия совместимости, записываемые в виде

$$\begin{cases} h_f \geq 0 \\ \dot{m}_{ice} \geq 0 \\ h_f \tilde{T} \geq 0 \\ \dot{m}_{ice} \tilde{T} \leq 0 \end{cases}$$
(3)

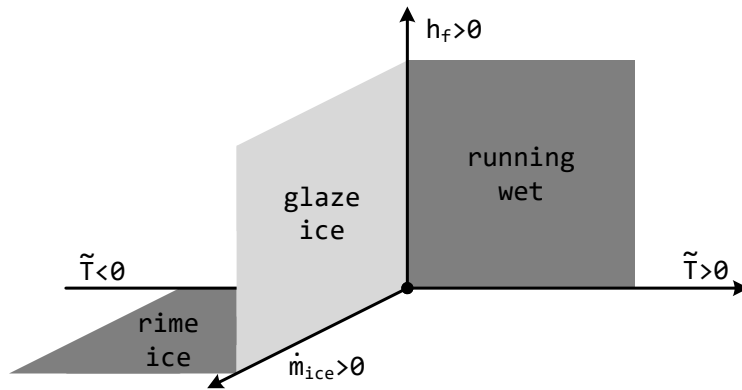


Figure 1. Пространство решение системы уравнений массового и теплового баланса ячейки.

Несмотря на то, что неизвестных в уравнении больше, чем самих уравнений, система все равно решается, так как данные переменные не являются полностью независимыми.

Решение ищется из условия того, что ячейка может находиться в одном из трех состояний. Первое состояние – running wet – достигается когда в ячейке полностью отсутствует лед и течет жидкая пленка. В этом случае температура не может быть отрицательной. Второе состояние – glaze icing – если в ячейке одновременно присутствует и лед, и вода. В этом случае температура равна нулю градусам по Цельсию. И наконец, в третьем случае при отрицательной температуре в ячейке не может находиться вода и присутствует только лед. Общее пространство решений показано на Fig. 2.

2. FEATURES OF THE EQUATIONS BEING SOLVED

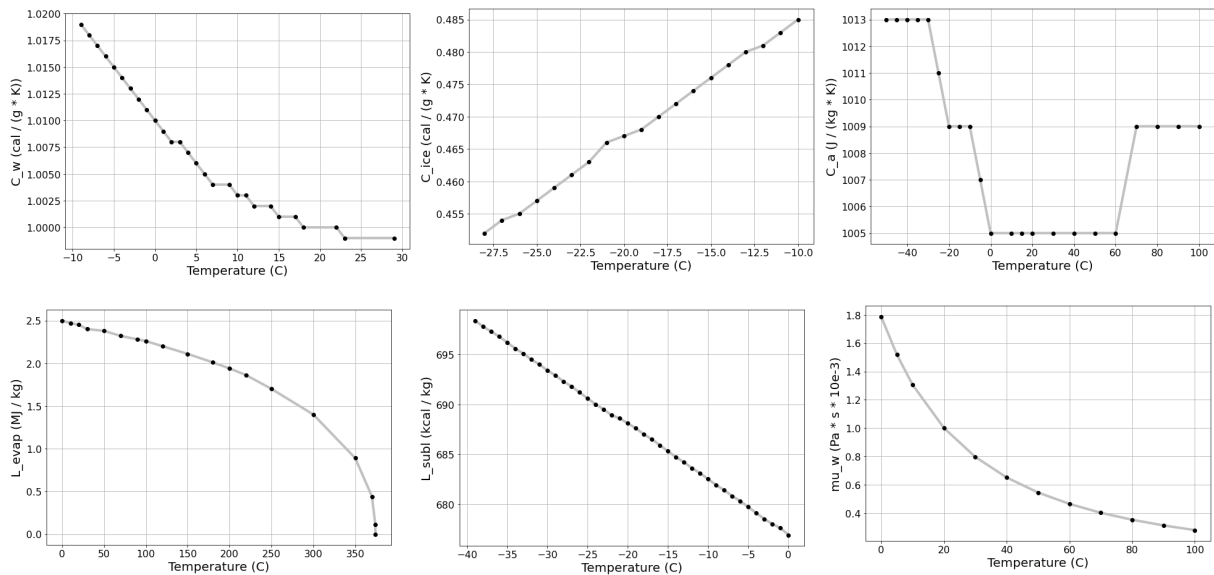


Figure 2. Зависимости физических величин от температуры.

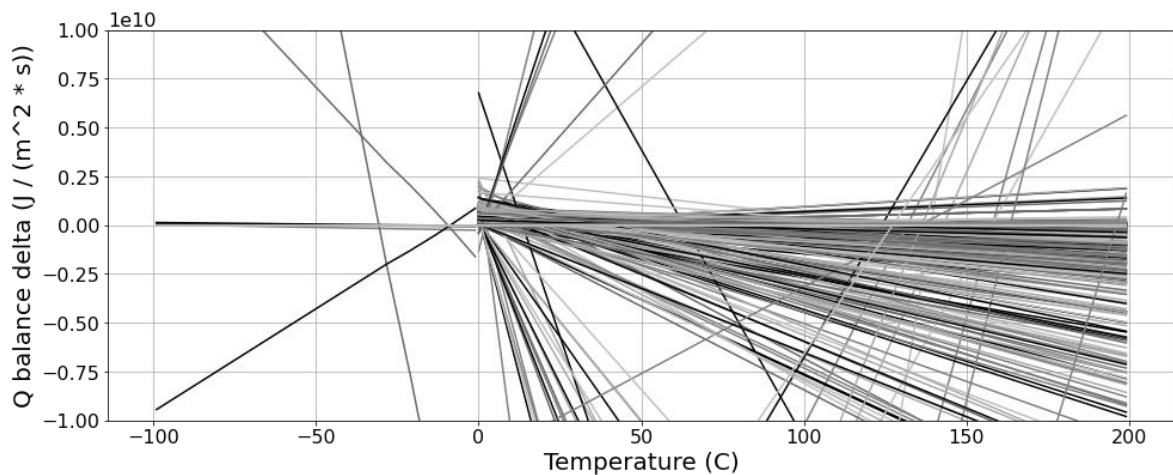


Figure 3. TODO

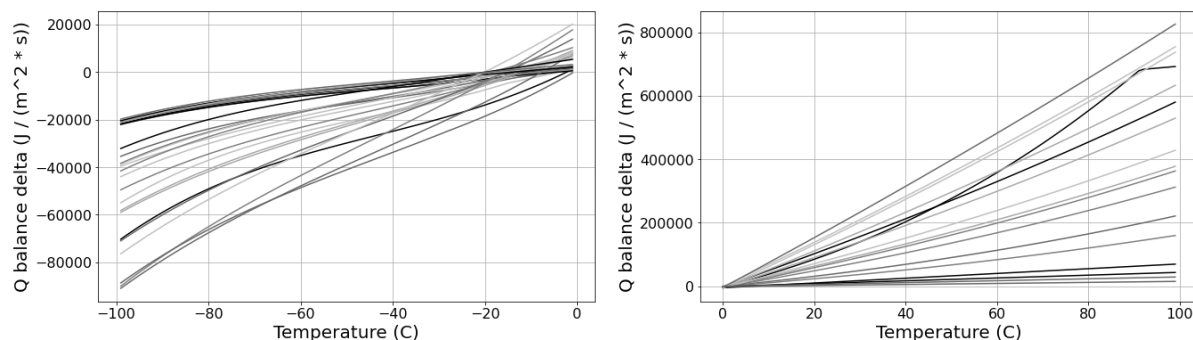


Figure 4. TODO

3. METHODS OF SOLVING NONLINEAR EQUATIONS

3.1. Метод деления пополам (метод бисекции)

Если известно, что на интервале (a, b) функция $f(x)$ имеет корень, для нахождения корня может быть применен метод деления пополам (метод бисекции). Его сущность заключается в следующем наборе шагов:

- Задать точность нахождения корня - ϵ .
- Разделить отрезок (a, b) пополам, получить два отрезка.
- Выбрать из двух получившихся в первом шаге отрезков отрезок, на концах которого выполняется условие $\text{sign}(f(a)) \neq \text{sign}(f(b))$.
- Если длина интервала превышает ϵ , повторить деление отрезка пополам. В противном случае корень находится в середине полученного интервала.

Для достижения заданной точности ϵ необходимо провести $\log_2 \frac{\epsilon_0}{\epsilon}$ операций, где ϵ_0 – длина начального интервала (a, b) , на котором находится корень; (a, b) – требуемая точность. Метод бисекции гарантированно найдет корень, если он существует. Если корней несколько, метод найдет один из корней.

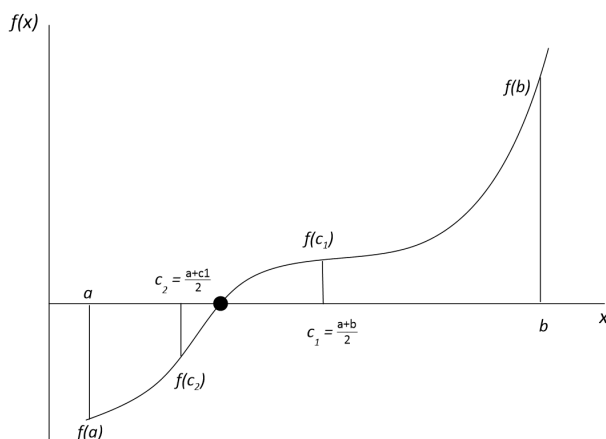


Figure 5. Метод деления пополам (метод бисекции)

3.2. Метод хорд

Метод хорд заключается в представлении функции $f(x)$ на отрезке (a, b) хордой, соединяющей концы интервала (a, b) . В результате точка пересечения хорды с $y = 0$ является искомым корнем уравнения. Если условие останова не удовлетворено $a - b < 0$, алгоритм повторяется на интервале (a, c) . Итерационная формула имеет вид:

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \quad (4)$$

Графическое представление метода хорд представлено на Fig. 6.

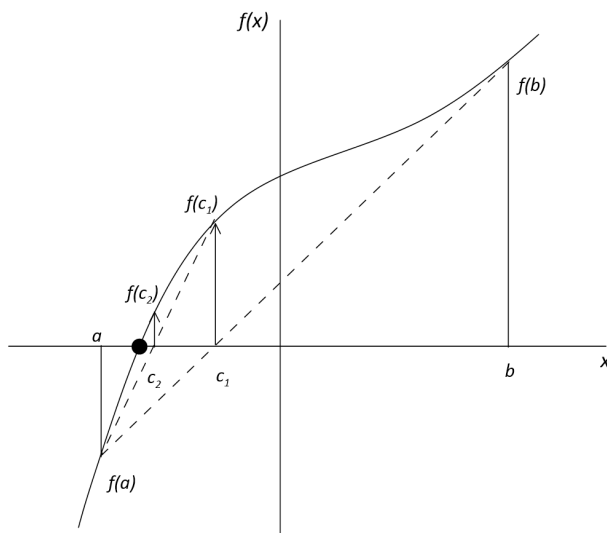


Figure 6. Метод хорд

3.3. Метод Ньютона

Данный метод отличается от вышеописанных тем, что он требует вычисления как функции $f(x)$, так и производной $f'(x)$ в произвольных точках x , принадлежащих интервалу (a, b) . Этот метод основан на разложении в ряд Тейлора в окрестности точки x :

$$f(x + \delta) \approx f(x) + f'(x)\delta + \frac{f''(x)}{2}\delta^2 + \dots \quad (5)$$

Для малых δ справедливо $\delta = -\frac{f(x)}{f'(x)}$. Таким образом, итерационная формула имеет вид:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (6)$$

Графическое представление метода Ньютона представлено на Fig. 7.

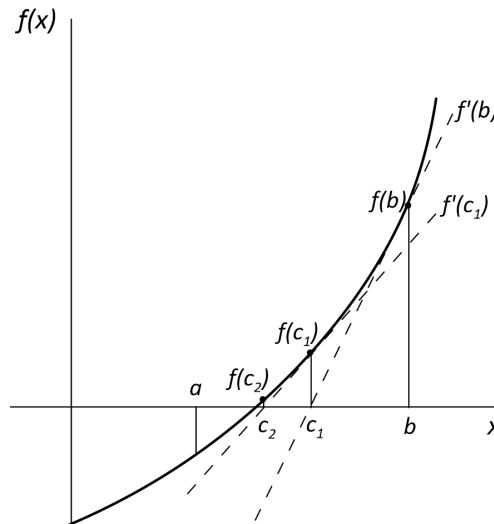


Figure 7. Метод Ньютона

3.4. Метод Брента

Метод Брента является модифицированным методом Деккера, который, в свою очередь, является объединением метода деления пополам с методом секущих. Сперва рассмотрим метод Деккера. Для каждой итерации метода рассматривается система

$$s = \begin{cases} b_k - \frac{b_k - b_{k-1}}{f(b_k) - f(b_{k-1})} f(b_k), & \text{if } f(b_k) \neq f(b_{k-1}). \\ m = \frac{a_k + b_k}{2} & \text{otherwise.} \end{cases} \quad (7)$$

Подразумевается, что на интервале (a, b) функция $f(x)$ имеет корень. Таким образом, на каждой итерации метода Деккера происходит выбор способа уточнения корня функции - при помощи метода деления пополам или же при помощи метода секущих. Слабая сторона метода Деккера - низкая скорость сходимости при малом $b_k - b_{k-1}$.

Брент предложил проводить дополнительную проверку $b_k - b_{k-1} > \delta$ для выбора метода уточнения корня, что позволяет повысить скорость сходимости метода. Кроме этого, линейная интерполяция в методе секущих была заменена обратной квадратичной интерполяцией, рекуррентное соотношение которой можно представить в виде

$$x_{n+1} = \frac{f_{n-1}f_n}{(f_{n-2} - f_{n-1})(f_{n-2} - f_n)}x_{n-2} + \frac{f_{n-2}f_n}{(f_{n-1} - f_{n-2})(f_{n-1} - f_n)}x_{n-1} + \frac{f_{n-2}f_{n-1}}{(f_n - f_{n-2})(f_n - f_{n-1})}x_n. \quad (8)$$

В данной формуле $f_n = f(x_n)$.

4. NUMERICAL EXPERIMENTS

Для анализа использования различных методов решения нелинейных уравнений были решены порядка 35000 уравнений, аналогичных уравнениям, показанным на Fig. 3.

	Failed, %	Avg. func calls	Avg. iterations
bisections	0.038	78.8	31.3
chords	0.038	11.2	3.7
newton	0.038	10.1	3.3
brenth	0.038	5.5	4.5

Принимая во внимание процент нерешенных уравнений, одинаковый для всех методов, можно сделать вывод, что эти уравнения не имеют решения на исследуемом интервале и это не связано с каким-либо методом. Распределение количества вызова функций и количества итераций представлено на Fig. 8 и Fig. 9 соответственно. Анализируя распределения, видно, что наиболее эффективным в смысле количества итераций и количества вызовов функции является метод Брента.

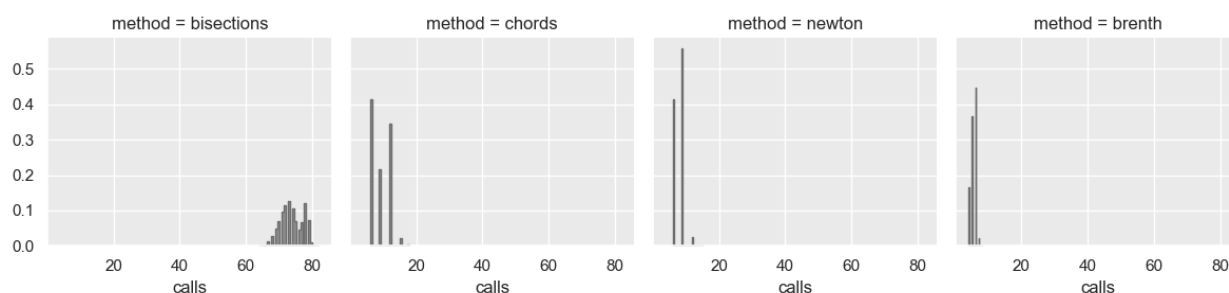


Figure 8. Распределение количества обращений к значениям функции для ее решения для исследуемых методов

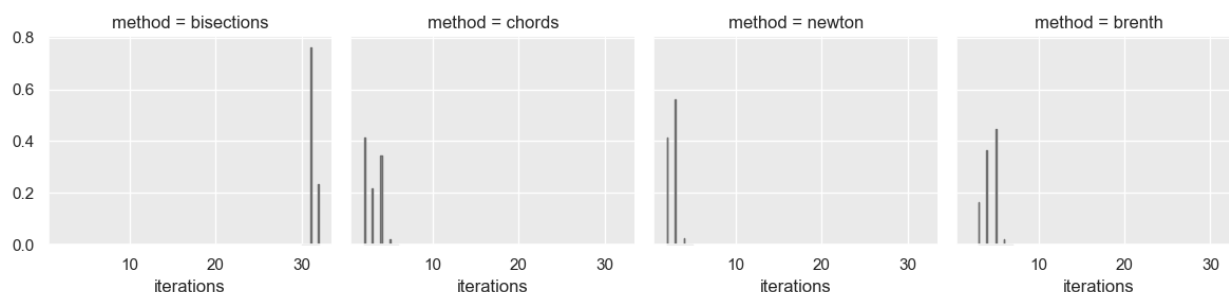


Figure 9. Распределение итераций для исследуемых методов

5. CONCLUSION

TODO

Acknowledgments. The work has been done at the JSCC RAS as part of the state assignment for the topic 0580-2021-0016. The supercomputer MVS-10P OP, located at the JSCC RAS, was used during the research.

REFERENCES

1. W. W. Wright, P. Struck, T. Bartkus, G. Addy, *“Recent Advances in the LIWICE Icing Model”*, SAE Technical Paper (2015).

2. Y. Bourgault, H. Beaugendre, W. G. Habashi, “*Development of a Shallow-Water Icing Model in FENSAP-ICE*”, Journal of Aircraft, Vol. 37, No. 4, 640–646 (2000).
3. H. Beaugendre, “*A PDE-Based Approach to In-Flight Ice Accretion*”, A thesis of the degree of Doctor of Philosophy, Department of Mechanical Engineering, McGill University, Montreal, Québec (2003).
4. Brent, R.P.. Finding a zero by means of successive linear interpolation. : London, 1969.
5. Brent, R.P.. Algorithms for minimization without derivatives. : Prentice-Hall, 1973.
6. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2007. Numerical Recipes 3rd Edition: The Art of Scientific Computing (3rd. ed.). Cambridge University Press, USA.