УДК. 371.6

РЕШЕНИЕ ЗАДАЧИ: ИЗ ТЕТРАДИ НА КОМПЬЮТЕР

Халиуллин Рауель Нигматзянович, к.п.н., доцент
rauel.haliullin@mail.ru
Маралов Омурбек, магистрант ОшГУ
maralov.Omurbek@bk.ru
Мамай уулу Женишбек, магистрант ОшГУ
jenishbeksalyev@gmail.com
Зарылбек уулу Артыкбек, магистрант ОшГУ
Artykbekzarylbekov@gmail.com
Ошский государственный университет,
Ош, Кыргызстан

Аннотация. *В статье рассматривается проблема разработки компьютерных моделей решения задач по физике. Перед учителями стоят две проблемы: 1) научить учащихся решению задач с помощью обобщенных алгоритмов решения задач и 2) научить учащихся завершать решение задачи численным экспериментом на основе компьютерной модели задачи. Дело в том, что одна и та же задача по-разному решается в тетради и на компьютере и пока не разработана методика, каким образом решение задачи с тетради переводится на компьютер. Авторами предлагается решение проблемы введением понятия «исполнительный алгоритм», выполняющего роль посредника между тетрадью по физике и компьютером.*

Ключевые слова. *Задача, методы решения, алгоритм, компьютерное моделирование, численный эксперимент, исполнительный алгоритм.*

ЧЕЧИЛГЕН МАСЕЛЕ: ДЕПТЕРДЕН КОМПЬЮТЕРГЕ

Халиуллин Рауель Нигматзянович, п.и.к., доцент
rauel.haliullin@mail.ru
Маралов Омурбек, ОшМУ магистрант
Maralov.Omurbek@bk.ru
Мамай уулу Женишбек, ОшМУ магистрант
jenishbeksalyev@gmail.com
Зарылбек уулу Артыкбек, ОшМУ магистрант
Artykbekzarylbekov@gmail.com
Ош мамлекеттик университети,
Ош, Кыргызстан

Аннотация. *Макалада физика боюнча маселелерди чечүү үчүн компьютердик моделдерди иштеп чыгуу маселеси каралат. Мугалимдердин алдында эки маселе турат: 1) окуучуларды маселелерди чыгаруунун жалпыланган алгоритмдерин колдонуу менен чыгарууга үйрөтүү жана 2) студенттерди маселенин компьютердик моделинин негизинде сандык эксперимент менен чыгарууну аягына чыгарууга үйрөтүү. Кептин баары бир эле тапшырма дептер менен компьютерде башкача чечилип, дептеден тапшырманын чечилишин компьютерге кантип өткөрүп берүү методикасы али иштелип чыга электигинде. Авторлор физикалык дептер менен компьютердин ортосунда ортомчу милдетин аткарган "аткаруучу алгоритм" түшүнүгүн киргизүү менен маселени чечүүнүн жолун сунушташат.*
Түйүндүү сөздөр. *Маселе, чечүү ыкмалары, алгоритм, компьютердик модель, сандык эксперимент, аткаруучу алгоритм*

SOLUTION OF THE TASK: FROM THE NOTEBOOK TO THE COMPUTER

Khaliullin Rauel Nigmatzyanovich, Ph.D., Associate Professor
rauel.haliullin @ mail. en
Maralov Omurbek, master student of Osh State University
Maralov.Omurbek @ bk. en
Mamai uulu Jenishbek, master student of Osh State University
jenishbeksalyev @ gmail. com
Zarylbek uulu Artykbek, master student of Osh State University
Artykbekzarylbekov @ gmail. com
Osh State University,
Osh, Kyrgyzstan

Abstract. *The article deals with the tasks of developing computer models for solving problems in physics. Teachers face two problems: 1) to teach students to solve problems using generalized task solving algorithms and 2) to teach students to complete the solution of a task with a numerical experiment based on a computer model of the task. The fact is that the same task is solved differently in a notebook and on a computer, and a methodology has not yet been developed for how the solution of a task from a notebook is transferred to a computer. The authors propose a solution to the problem by introducing the concept of "executive algorithm", which acts as an intermediary between a physics notebook and a computer.*
Keywords. *The task, solution methods, algorithm, computer simulation, numerical experiment, executive algorithm.*

Введение. Решение задач в процессе обучения физике играет особо важную роль: во-первых, не только закрепление и углубление полученных знаний, но, главное, - это развитие умственных способностей учащихся. Применение компьютеров в процессе решения задач не ставит целью облегчить и ускорить вычисления, а путем компьютерного моделирования задач, научить учащихся ставить эксперименты, глубже исследовать физические явления и закономерности в природе. Но для этого важно научить учащихся решать задачи. В последнее время интерес к проблемам обучения учащихся решению физических задач традиционными методами связан с использованием обобщенных алгоритмов решения задач, которые могут служить путеводителем и ориентиром для учащихся в самостоятельном решении задач [3; 5]. Поэтому оптимальным выходом, позволяющим добиться эффективного образовательного результата на уроках решения задач становится компьютерное моделирование решения задач.

Обсуждение и результаты. Разработка алгоритмов работы компьютерных моделей позволяет не только сформировать навык традиционного решения физических задач, но и способствовать освоению и применению компьютерных умений в работе с компьютерными моделями [1; 4]. При этом, работа с математической моделью задачи тесно связывается с разработкой соответствующего алгоритма для соответствующей компьютерной модели задачи. Решить задачу – значит, согласовать теорию с практикой. Мысленный эксперимент с математической моделью задачи, подкрепленный вычислениями. В процессе решения задач знания учащихся конкретизируются, создаётся понимание сущности явлений, физические понятия и величины приобретают реальный смысл, у ученика развивается способность рассуждать, устанавливать причинно-следственные связи, выделять главное и отбрасывать несущественное. Решение задач позволяет сделать знания осознанными, избавить от формализма, заучивания законов и формул. Это - образовательная функция решения задач.

Решить задачу – не значит только получение ответа на вопрос, поставленный в задаче, а возможность всесторонне и глубоко изучить физическое явление, затронутое в задаче. А для убедительности и достоверности результата требуется реальный эксперимент, который можно заменить численным экспериментом на основе компьютерных технологий.

В последние годы все большее внимание методистов привлекает проблема применения алгоритмов для обучения учащихся решению физических задач [2; 5]. Разрабатываются примерные алгоритмы решения задач по различным разделам физики, но они не касаются непосредственно компьютерного моделирования решения задач. Именно здесь возникают проблемы, связанные с алгоритмами работы компьютерных моделей. Все знают, что работа любой компьютерной программы основывается на алгоритмах. Разработать компьютерную модель, значит разработать соответствующий алгоритм. Алгоритм вводится в компьютер на соответствующем языке программирования. Поэтому алгоритм создается в качестве основы программирования. Сначала алгоритм – потом программа.

Для создания компьютерной программы требуется обеспечить уверенный переход от решенной в тетради задачи к разработке ее компьютерной модели. При этом, компьютер в решении задач применяется не как средство вычислений, например, в качестве калькулятора, а как средство проведения виртуального физического эксперимента по содержанию конкретной задачи.

Алгоритм – описание последовательности действий (план), строгое выполнение которых приводит к достижению поставленной цели за конечное число шагов [1]. Решение любой задачи состоит из определенной последовательности действий – этапов решения задачи. Нельзя придумать алгоритм, пригодный для решения любой задачи. Каждая задача решается по своему неповторимому алгоритму.

С другой стороны, решение задач по различным разделам физики также резко отличается от остальных. Но, есть операции, повторяющиеся при решении задач конкретного раздела физики

Например, для решения задач на второй закон Ньютона методисты предлагают алгоритм, представляющий последовательность предписаний, например, алгоритм решения задач по динамике материальной точки [2]:

1. Выбрать систему и точку отсчета.
2. Создать рисунок, изобразить все силы, действующие на тело. Указать направления ускорений.
3. Записать закон Ньютона в векторной форме и через проекции перейти к скалярной записи.
4. Выразить все силы через величины, от которых они зависят.
5. При необходимости к полученным уравнениям динамики добавить кинематические уравнения.
6. Решить полученную систему уравнений и провести вычисления.

Этот примерный алгоритм нацелен на решение задачи в тетради, но с компьютерным моделированием не связан.

Для чего предназначена тетрадь по физике? Для контроля учителем самостоятельной работы ученика? С точки зрения методистов – это поле для самостоятельной работы, это обучение решению задач, где испытывается настойчивость, трудолюбие, сознательность, развивается мышление, расширяются знания, глубже усваивается учебный материал [3].

А что собой представляет решение задачи в тетради? Это применение на практике накопленных знаний и умений, расширение имеющихся знаний, а в отдельных случаях восполнение пробелов в знаниях. Учащиеся решают задачи в классе и дома. В классе задачи решаются одновременно у доски и в тетради, а дома - в тетради, без посторонней помощи. В любом случае в тетради сохраняются следы, умственной деятельности ученика, связанные с процессом решения задачи.

А для чего решать задачу на компьютере, если она уже решена в тетради? При этом математическая модель решенной задачи преобразуется в компьютерную модель, а целью создания модели, мы считаем, – эксперимент с моделью. Как, например, реальные эксперименты с моделью самолета, моделью ракеты и так далее. Прежде всего, надо выяснить, что можно найти в тетради, и что нужно взять из нее, чтобы перенести на компьютер.

Для полной ясности необходимо, в первую очередь, ознакомиться с текстом содержания задачи, но его, как правило, в тетради нет: его можно найти в задачнике по номеру задачи. А теперь посмотрим, что в тетради? Номер задачи, дано, найти, затем рисунок, куча формул и, в конце, расчетная формула (математическая модель задачи) и ответ. Процесс вычислений, как правило, остается в черновике. Вот и все.

Иногда и процесс преобразования формул и уравнений вместе с вычислениями также остается в черновике, а в тетрадь заносятся чистые формулы и числовые результаты. Как все это перенести на компьютер, то есть построить компьютерную модель задачи? Для этого нужен алгоритм решения данной задачи, а его в тетради нет. Алгоритм для компьютерной модели нужно разрабатывать, пока решение задачи сохраняется в сознании. Разрабатывать такой алгоритм так же сложно, как и решение задачи. Чем труднее задача, тем сложнее алгоритм. Значит, алгоритм адекватной компьютерной модели, каким–то образом, должен быть отражен в тетради. Этот, пока невидимый, алгоритм мы назвали исполнительным алгоритмом.

Для простых, одноходовых задач ничего не нужно предпринимать: алгоритм - на виду: это исходные данные задачи и рабочая (расчетная) формула. Исходные данные, отраженные в тетради, абсолютно готовы для компьютерной модели. Дело в том, что все числовые данные приведены в единую систему SI. Для исполнительного алгоритма нужна только вычислительная часть. Сколько формул, столько и шагов, но некоторые формулы могут использоваться многократно.

Когда учащиеся готовятся к контрольной работе или к экзаменам, учителя предлагают им просмотреть ранее решенные задачи. Они перелистывают свои тетради, бегло просматривают записи решений, не всегда четко представляя содержание задачи по записям «Дано» и «Найти».

Чтобы облегчить вникание в содержание задачи, помогают рисунки, графики, выполненные в процессе решения. Поскольку каждая задача связана с конкретным физическим явлением или событием, мы предлагаем придумывать уникальный заголовок к задаче: в дальнейшем заголовок вместе с рисунком помогает легко воспроизвести смысл и содержание давно решенной задачи. В дальнейшем, это же название послужит заголовком для компьютерной модели.

При просмотре ранее решенных задач, учащийся пытается мысленно восстановить смысл и содержание задачи. В этом ему помогают рисунки, схемы, графики, сделанные в ходе решения задачи. Этот этап решения задачи мы называем идеализацией и визуализацией. Вместе с заголовком рисунки помогают быстрее понять содержание задачи и ход ее решения.

Если учащийся пытается понять содержание и решение задачи в чужой тетради, его ждут большие трудности (чужие мысли не прочтешь). Но если в ней имеются признаки исполнительного алгоритма, все становится на места, если компьютер понимает, а человек, - тем более. В результате решения системы уравнений часто получается формула, которой нет в учебниках. При повторении решенных задач по тетрадям такая формула узнается не сразу. Это расчетная формула.

Решение задачи сопровождается логическими операциями, рассуждениями, в которых участвуют факты, события, отношения, связи объектов. В методике решения задач различают два противоположных метода решения: аналитический и синтетический.

Синтетический метод напоминает решение следственной задачи, которое начинается с имеющихся у следствия улик. Каждый шаг следователя опирается на известные факты и данные, и с каждым шагом вскрываются новые факты (новые данные для следующего шага), …, и так, шаг за шагом, пока задача не будет решена окончательно.

Аналитический метод напоминает следствие, начинающееся с гипотезы, под которую подгоняются имеющиеся улики и факты. В качестве гипотезы принимаются требования задачи. В этом случае, если гипотеза не подтверждается, выдвигается следующая гипотеза.

Аналитический метод имеет положительные и отрицательные стороны. Совершенствуются математические умения создавать и преобразовывать формулы, решать системы уравнений, экономится время на изнурительных вычислениях. Расчетная формула избавляет от излишних вычислений, экономит время, ответ, как говорится, прямо «просится в руки».

Недостаток метода в том, что полученная в результате математических преобразований, расчетная формула не всегда имеет чисто физический смысл и ее трудно понять с первого взгляда.

При традиционном решении задач в большинстве случаев используется метод анализа: решение начинают непосредственно с требования задачи. То есть, ищут формулу, из которой можно выразить искомую величину. При этом в первой же формуле присутствуют величины, значения которых в условиях задачи не оговорены, но их можно выразить через другие величины, так продолжается, пока не будет сформирована формула. в которой представлены все данные из текста задачи и справочных таблиц.

Метод анализа начинается непосредственно с требования задачи, метод синтеза начинается с исходных данных. Соответственно от неизвестного – к известному и от известного – к неизвестному.

Одну и ту же задачу можно решить аналитическим и синтетическим методом. Рассмотрим, например, решение простой задачи:

Задача 1. Требуется вычислить давление на грунт типового строительного кирпича с габаритными размерами 25 см×12см×6,5см. Плотность материала кирпича ρ известна

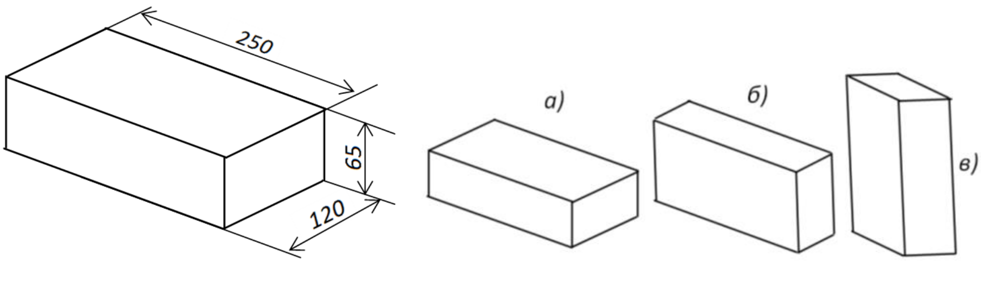


Рис.1. Размеры стандартного строительного кирпича

У кирпича три измерения (ширина - d, длина - l и толщина – h). Дано: d, l, h, g, ρ, найти: $p_{мин}$. Решение задачи заключается в сравнении давления кирпича на грунт при трех положениях кирпича (рис. 1). Рассмотрим последовательность решения задачи для первого варианта положения кирпича обоими методами (рис. 1 а).

1. Последовательность решения задачи *аналитическим* методом:

$$p = \frac{F}{S} \;\rightarrow\; F = m \cdot g \;\rightarrow\; m = \rho \cdot V \;\rightarrow\; V = S \cdot h \;\rightarrow\; S = d \cdot l$$

Подставляя выражения для F, m, V, S в формулу $p = \frac{F}{S}$, получим

$$p = \frac{F}{S} = \frac{m \cdot g}{d \cdot l} = \frac{\rho \cdot V \cdot g}{d \cdot l} = \frac{\rho \cdot S \cdot h \cdot g}{d \cdot l} = \frac{\rho \cdot d \cdot l \cdot h \cdot g}{d \cdot l} = \rho \cdot g \cdot h \qquad (1)$$

получаем расчетную формулу: $p = \rho \cdot g \cdot h$, которая очень похожа на формулу подсчета давления в жидкостях с плотностью ρ на глубине h.

Формула показывает, что даже давление кирпичной стены на грунт не зависит от площади основания стены, а только от толщины кладки. Это следует из того, что значения величин l = 25 см и d = 12 см в данном случае оказались ненужными.

Но исполнительный алгоритм для эксперимента будет использовать все исходные данные. И, если бы в задаче не были указаны габариты кирпича, а только высота h (задача с неполными данными), учащиеся отказались бы решать такую задачу. Без габаритных размеров массу кирпича найти невозможно. В подобных случаях при компьютерном моделировании задач учащимся можно предложить самим вводить разумные значения для недостающих величин.

При других положениях кирпича также будут получены расчетные формулы, в которых давление определяется только высотой:

$$p = \rho \cdot g \cdot d \qquad (2)$$

$$p = \rho \cdot g \cdot l \qquad (3)$$

Из сравнения результатов вычислений делаются два вывода:

а) минимальное давление, когда кирпич на поверхности лежит плашмя (площадь опоры больше);

б) давление не зависит от площади опоры, а зависит только от высоты кирпича над поверхностью.

2. Последовательность решения этой же задачи *синтетическим методом*:

$$S = a \cdot b \;\rightarrow\; V = S \cdot h \;\rightarrow\; m = \rho \cdot V \;\rightarrow\; F = m \cdot g \;\rightarrow\; p = \frac{F}{S}.$$

Решение заканчивается знакомой формулой давления $p = \frac{F}{S}$. При этом в решении задачи участвуют все исходные данные. Прочитав данный алгоритм, ученик не сделает вывод, что давление не зависит от площади, а только от высоты. Поэтому учащимся предлагается вычислить давление для трех положений кирпича и только потом, на основании результатов, делать соответствующие выводы о кирпичной стене.

Чтобы увидеть, как отличаются решения данной задачи методом анализа и методом синтеза, расположим параллельно обе последовательности формул:

$$p = \frac{F}{S} \;\rightarrow\; \mathrm{F} = m \cdot g \;\rightarrow\; m = \rho \cdot \mathrm{V} \;\rightarrow\; \mathrm{V} = \mathrm{S} \cdot \mathrm{h} \;\rightarrow\; \mathrm{S} = \mathrm{a} \cdot \mathrm{b} \qquad (1)$$

$$S = a \cdot b \;\rightarrow\; V = S \cdot h \;\rightarrow\; m = \rho \cdot V \;\rightarrow\; F = m \cdot g \;\rightarrow\; p = \frac{F}{S} \qquad (2)$$

Как видим, формулы выстроились в обратном порядке, таким образом, что создание исполнительного алгоритма возможно при любом из методов решения задачи.

Иногда решить задачу методом синтеза невозможно, это задачи с неполными данными. В процессе математических преобразований отдельные величины сокращаются, и их в расчетной формуле нет. Чтобы, «заставить» компьютер решать подобные задачи, рекомендуется вводить эти недостающие величины и, чтобы не «насмешить» проверяющих, вводить вполне правдоподобные значения.

Компьютер не умеет преобразовывать формулы, например, выразить x из уравнения $ax + в = c$, или находить неизвестные величины из системы уравнений. Формулы и уравнения для вычислений вводятся в компьютер в готовом к вычислению виде. Математическая модель задачи, рабочая (расчетная) формула, принятая за основу для вычислений, не всегда отражает сущность соответствующего физического закона. Расчетные формулы необычны тем, что в учебниках они, как правило, отсутствуют – это результат математических преобразований при решении системы уравнений.

Если задача сложная, она содержит несколько расчетных формул, так или иначе, связанных друг с другом. При этом исполнительный алгоритм может содержать ветвления, циклы, подпрограммы, нестандартные функции.

А если в задаче требуется найти ответ на несколько вопросов, соответственно число расчетных формул увеличивается. Расчетная формула – часть математической модели задачи, по своему виду не всегда совпадает с физической моделью явления.

Если раньше нужно было просто решить и проанализировать решение задачи, теперь эту же задачу нужно решить заново, на компьютере. Считается, что, если учащийся решил задачу, значит, он знает соответствующий учебный материал. Но решение задачи на компьютере означает не только применение и закрепление имеющихся знаний, но также их расширение и углубление.

Итак, задача решена, теперь решение нужно перенести на компьютер. Что означает «компьютерное моделирование решения задачи»? Учащийся сам полностью решает задачу и затем «учит» компьютер повторно решить эту же задачу с возможностью, оперативно меняя данные, организовать физический эксперимент, более глубоко исследовать физическое явление, лежащее в основе содержания задачи.

Успешному созданию компьютерной модели способствует умение «думать за машину», ставить себя на место компьютера. Компьютерная модель позволяет всесторонне исследовать физическое явление, отраженное в содержании задачи. Компьютерная модель позволяет многократно решить одну и ту же задачу при других исходных данных, то есть, всесторонне исследовать физическое явление, описанное в задаче. Чтобы можно было наблюдать это явление, компьютер должен отразить его на экране в виде текста, символов, таблиц, графиков, рисунков, анимации.

Непосредственно от тетради сразу к компьютеру перейти невозможно, нужно разрабатывать соответствующую программу для компьютера. Чтобы преобразовать решение в программу для компьютера, нужен переводчик, посредник. Эту роль мы возлагаем на исполнительный алгоритм.

Исполнительный алгоритм – мостик от математической модели задачи к компьютерной модели, от тетради - к компьютеру. Необходимые качества исполнительного алгоритма – краткость, наглядность, понятность, привлекательность, впечатлительность, убедительность, изящность, ...

Не для каждой задачи требуется разработка исполнительного алгоритма. Для чего разрабатывать компьютерную модель, если задача аналитически решена и осталось только вычислить результат. Это можно осуществить с помощью калькулятора или ручных вычислений с карандашом. Создание компьютерной модели оправдано для многоходовых, логически сложных, иногда противоречивых задач. Причем, с предоставлением возможности проведения численного эксперимента, повышения наглядности и убедительности получаемых результатов.

Компьютерные модели простых задач представляют программы линейной структуры, без логических переходов. При этом линейная программа состоит из трех частей: 1) ввод данных, 2) вычисление, 3) вывод результата на экран. Исполнительный алгоритм отвечает только за вычислительную часть программы. Для многоходовых и сложных задач исполнительный алгоритм можно представить в виде сгруппированных по определенному признаку групп формул. Их можно записать отдельно, в конце решенной задачи, или пронумеровать имеющиеся уравнения или последовательно соединить стрелками (рис. 2 б). Таким образом, в тетради отражаются признаки исполнительного алгоритма.

1) $S = a \cdot b$
2) $V = S \cdot h$
3) $m = \rho \cdot V$
4) $F = m \cdot g$
5) $p = \frac{F}{S}$

$$S = a \cdot b \rightarrow V = S \cdot h \rightarrow m = \rho \cdot V \rightarrow F = m \cdot g \rightarrow p = \frac{F}{S}$$

Рис. 2. Символьное и графическое представление исполнительного алгоритма

Для простой задачи, где в решении всего одна-две расчетные формулы, исполнительный алгоритм практически не нужен: в этих случаях разработка программы не представляет особой сложности. Просматривая исполнительный алгоритм, предназначенный для компьютерной модели, учащийся мысленно, шаг за шагом, в своем воображении, воспроизводит процесс решения задачи моделью. Но, если что-то непонятно, он может с самого начала прочитать свои записи, сделанные в тетради в процессе аналитического решения задачи. А вот с исполнительным алгоритмом все просто: последовательность легко узнаваемых формул подсказывает логику решения задачи, причем каждая формула представляет прямую или модифицированную запись формулы определенного закона физики, например,

$$F = ma \quad \text{или} \quad a = \frac{F}{m} \quad \text{или} \quad m = \frac{F}{a}$$

Учащийся сразу догадывается, что это второй закон Ньютона.

Для создания исполнительного алгоритма, то есть, для установления порядка работы компьютерной модели задачи, надо ясно представлять логику решения задачи: что, откуда, как, для чего и так далее.

Исполнительный алгоритм компьютерной модели, представляющий цепочку формул, основан на решении задачи методом синтеза: при этом каждая предыдущая формула алгоритма готовит данные для следующей формулы и на каждом этапе вычисления проводятся над формулами с полным набором данных. При этом возрастает доля вычислительной работы, но для компьютера, с его колоссальной скоростью вычислений, это не сложно.

Таким образом, при создании исполнительного алгоритма учащийся должен поставить себя на место компьютера, то есть, последовательно решать ту же задачу, опираясь на каждом шагу только на известные данные и на каждом шагу готовить информацию для следующего шага. И так до получения окончательного результата.

Что дает создание и разработка исполнительного алгоритма для ученика?

1) Формирует умение планировать эксперимент, руководить.

2) Обеспечивает прямой перевод задачи в компьютерную модель

3) Ускоряет понимание при повторении ранее решенной задачи

Мы считаем, что создавать исполнительный алгоритм необходимо даже независимо от того, будет ли разрабатываться компьютерная модель. Решив задачу, тут же, в тетради зафиксировать или записать (зарисовать) исполнительный алгоритм в виде упорядоченной последовательности формул, структурных схем. Последовательность формул удобна для компьютерного моделирования задачи с помощью электронных таблиц (Excel).

В чем главная особенность исполнительного алгоритма? Фактически, это программа работы компьютерной модели, написанная на понятном языке, доступная любому ученику в классе, любому учителю физики или математики, даже человеку, не знакомому с основами программирования.

Рассмотрим решение и составление исполнительного алгоритма задачи по кинематике материальной точки.

Задача 2. С высоты $h_1 = 10$ м брошен камень, горизонтально со скоростью v = 5 м/с. Одновременно с высоты $h_2 = 5$ м под углом брошен другой камень. С какой начальной скоростью и под каким углом брошен второй камень, если камни встретились на высоте $h = 1$ м над землей?

Так как камни стартуют одновременно и встречаются в одной точке, это означает, что горизонтальные составляющие скоростей равны, то есть $v_{1x} = v_{2x} = v$. Необходимо узнать вертикальную составляющую v_{2yo}.

Проекции скорости первого камня: $v_{1xo} = v$, $v_{1yo} = 0$. Проекция $v_{2xo} = v$, следовательно, задачу можно решать, скажем, выбрав относительную систему координат, связанную со вторым камнем.

В совместном полете первый камень все время находится над вторым и постепенно расстояние между ними сокращается.

Первый камень до встречи со вторым опускается вниз на $h_1 - h$ = 9 м. Время его падения до высоты h = 1 м определим из формулы:

$$h_1 - h = \frac{gt^2}{2}, \quad \text{откуда} \quad t = \sqrt{\frac{2(h_1 - h)}{g}} \qquad (1)$$

Уравнение движения второго камня имеет вид: $y = h_2 + v_{2yo} \cdot t - \frac{gt^2}{2}$. Тело вначале поднимается вверх с начальной скоростью v_{2yo}, достигает некоторой высоты и начинает опускаться вниз. С учетом (1) и $y = h$ запишем

$$h_2 - h = v_{2yo} \cdot t - \frac{gt^2}{2},$$

откуда начальная горизонтальная составляющая скорос ти второго камня равна

$$v_{2yo} = \frac{h_2 - h}{t} + \frac{gt}{2} \qquad (2)$$

Теперь найдем требуемую начальную скорость второго камня

$$v_2 = \sqrt{v^2 + v_{2yo}^2} \qquad (3)$$

Угол, под которым нужно бросить, чтобы камни встретились в нужной точке на высоте h:

$$\alpha = arctg\frac{v_{2yo}}{v} \qquad (4)$$

Вычисления дают следующие результаты:

t = 1, 28 сек; v_{2yo} = 3,91 м/с; v_2 = 6,35 м/с; α = 38, 06°.

Исполнительный алгоритм компьютерной модели отдельно составлять нет необходимости: это последовательность из четырех уравнений (1), (2), (3), (4). С помощью компьютерной модели можно провести эксперимент с пересечением траекторий на высотах от h = 0, 1, 2, 3, … почти до 10 м!

Можно также построить график зависимости скорости от требуемой высоты места встречи, ввести ограничение скорости, например $v_{2макс} = 2v$ и т.д.

Ниже приведен фрагмент программы компьютерной модели, в которой решение задачи сопровождается построением графика совместного полета камней (рис. 3).

Исполнительный алгоритм может представляться в трех видах (словесное, графическое (структурные схемы, блок-схемы), аналитическое).

Словесный способ описания алгоритма представляет собой описание последовательных этапов обработки данных и задается в произвольном изложении на естественном языке. Словесный исполнительный алгоритм решения задачи можно увидеть у учителя, в конспекте открытого урока по решению физических задач. Но, если записать все разъяснения и комментарии учителя, демонстрирующего решение задачи у доски – это и будет основой исполнительного алгоритма (см. аналитическую часть решения нашей задачи, где разъяснено, что, откуда взять, куда ввести, что получится при этом).

```
. . . . . . . . . . . .
V1Y0 = 0: V1X0 = V: V2X0 = V
VMAX = 9.6
T0 = SQR (2 * (H1 - H0) / G)
PRINT "TO = "; T0
VY20 = (H0 - H2) / T0 + G * T0 / 2
PRINT "VY20 = "; VY20
V2 = SQR (VY20 ^ 2 + V ^ 2)
PRINT "V2 = "; V2
ALFA =ATN(VY20/V)*180/3.14
PRINT "ALFA = "; ALFA;" GRAD"
TM = SQR (2 * H1 / G)
PRINT " TM = "; TM
TC = T1 * 100: C = 11
FOR T = 0 TO TM STEP .05
   Команды построения
   графиков поточкам
NEXT T
END
```

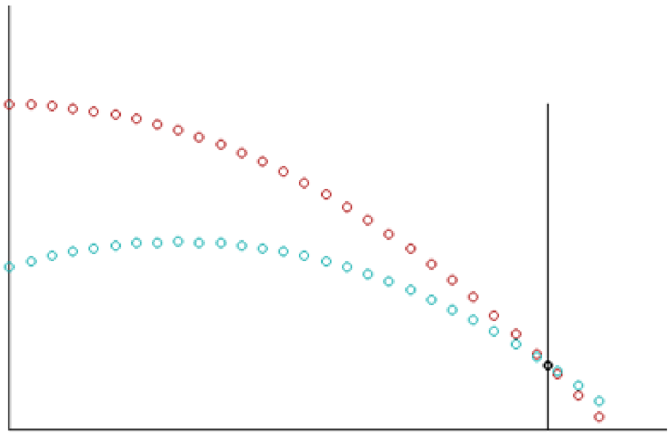


Рис. 3. Компьютерная модель задачи «Два камня»

Достоинством данного способа является простота описания, а к недостаткам отнести то, что такой подход многословен и не имеет строгой формализации, поэтому допускает неоднозначность толкования отдельных предписаний, в силу чего словесный способ представления алгоритма не получит широкого распространения.

Графический способ представления исполнительного алгоритма. Наиболее полноценное графическое представление алгоритма – это блок-схема. Это - графическое представление хода решения задачи с помощью компьютера. Блок-схема состоит из блоков, соединенных линиями, а блоки изображаются в виде геометрических фигур, называемых символами. Внутри символов записываются указания о выполняемых блоком функциях: формулы, текст, логические выражения. Хотя блок-схема своей наглядностью создает условия для ясного понимания структуры и логики компьютерной модели, однако по ней невозможно детально проследить работу модели, так как в блоках невозможно разместить все используемые формулы и уравнения (ограниченность площади блоков). В этом виде исполнительный алгоритм отражает все возможные вычислительные шаги компьютерной модели, от начала и до конца. Не обязательно выражать исполнительный алгоритм каждой задачи блок-схемой. Это необходимо только для очень сложных задач, связанных с многократными вычислениями.

С целью экономии времени при разработке исполнительного алгоритмы можно ограничиться только нумерацией расчетных формул, но при этом алгоритм теряет свою наглядность и понятность. Но это необходимо, чтобы принцип работы компьютерной модели был понятен и доступен для понимания работы модели.

Задачи с недостающими или избыточными данными. В первом случае. имеют место случаи с задачами в которых в процессе математических преобразований происходит сокращение некоторых величин, и как следствие, в расчетной формуле их нет. Но для работы компьютерной модели эти величины необходимы, и если их в условии нет, в таких случаях можно, зная, что это не повлияет на результат, вводить разумные значения недостающих величин, например, не имеет смысла растягивать пружину на метры, или брать диаметр мяча в миллиметрах.

Видя, что в процессе решения уравнений некоторые величины взаимно сократились, а среди данных в тексте задачи их нет, при создании исполнительного алгоритма для компьютерной модели необходимо ввести произвольные значения этих величин, зная, что в процессе вычислений они все равно будут аннулированы. Это касается также задач с неполными данными.

С подобными задачами учащиеся сталкивались не раз. Например:

Задача 3. Под каким углом надо бросить тело, чтобы дальность его полета была равна высоте полета?

Дано: S = h; v=10. Найти: α. Вот и вся задача.

В задаче ничего не дано, а ответ требуется получить конкретный, в цифрах. Здесь нет числовых данных, но задачу можно решить с помощью математических преобразований формул и уравнений, либо начальную скорость учащийся выбирает произвольно, полагаясь на свой опыт. Конечно, он не возьмет скорость муравья или сверхзвукового самолета.

Мы не приводим решения задачи. Эта задача решается в тетради аналитическим или синтетическим методом через цепочку уравнений:

$$v_x = v \cdot cos\alpha; \quad v_y = v \cdot sin\alpha; \quad t = \frac{v_y}{g}; \quad h = \frac{gt^2}{2} = \frac{v_y^2}{2g}; \quad S = v_x \cdot t = \frac{v_x \cdot v_y}{g}$$

$$\mathrm{S} = \mathrm{h}; \quad \frac{v_x \cdot v_y}{g} = \frac{v_y^2}{2g} \quad 2v_x = v_y; \quad tg\alpha = \frac{v_y}{v_x} = 2; \quad \alpha = arctg2 \approx 63{,}5^0.$$

Все это решается в тетради, а что для модели? А на модели можно организовать эксперимент: при заданном значении начальной скорости поиск значения угла α, при котором $S = S_{макс}$. Это сложная задача.

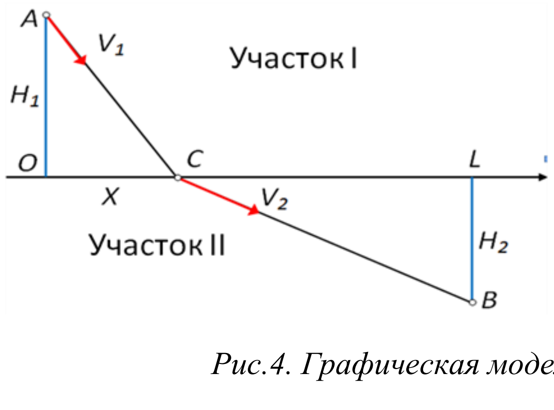
Как «научить» компьютер решать сложные задачи? К каждой задаче нужен свой, неповторимый, алгоритм. Алгоритм сложной задачи содержит ветвления, циклы, подпрограммы, нестандартные функции. Кроме того, компьютер должен сообщать результаты решения текстом, числами, графиками, диаграммами, схемами, таблицами. И всему этому надо «научить», чтобы компьютер выдал информацию в наиболее убедительной, доступной и наглядной форме.

Для реализации компьютерной модели нами используется простейший и доступный язык программирования Qbasic 64. Это легко читаемый и понимаемый учащимися язык. Программы с этого языка легко перевести на любые другие более сложные языки программирования.

Рассмотрим задачу на принцип Ферма. В оптике при изучении законов распространения света упоминается принцип Ферма, заключающийся в том, что свет от источника к приемнику света распространяется кратчайшим путем (за минимум времени). Решение такие задач связано с нахождением максимумов и минимумов исследуемых функций. Рассмотрим такую задачу.

Задача 4. Курьер должен доставить почту из пункта A в пункт B, за минимальное время. В какой точке X следует пересечь границу раздела двух участков, если известно, что скорость курьера в первой зоне 5 км/ч, а во второй зоне 3 км/ч. Расстояния: L = 1000 м, H1 = 300 м, H2 = 200 м.

Из рисунка видно, что минимум времени курьер затратит, пересекая линию OL где-то в промежутке между точками O и L.



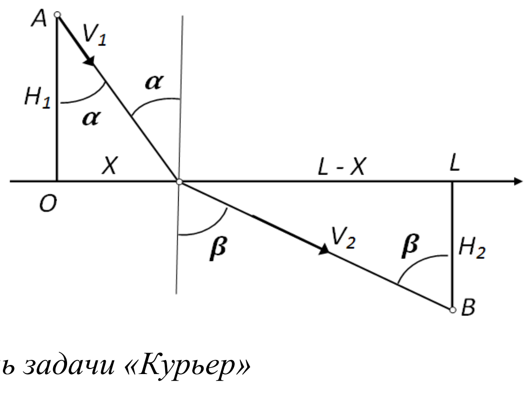


Рис.4. Графическая модель задачи «Курьер»

Решение: Курьер совершает переход из пункта A в пункт B. Пути AC = S1 и CB = S2 он проходит с постоянной скоростью v1 и v2. Из рисунка 4 видно, что $sin\alpha = \frac{x}{S_1}$; $sin\beta = \frac{L-x}{S_2}$. Учитывая, что $S_1 = v_1 t_1$ и $S_2 = v_2 t_2$, получим:

$$sin\alpha = \frac{x}{v_1 t_1} \quad \text{и} \quad sin\beta = \frac{L-x}{v_2 t_2}.$$

Отсюда $t_1 = \frac{x}{v_1 sin\alpha}$ и $t_2 = \frac{L-x}{v_2 sin\beta}$

Так как общее время $t = t_1 + t_2$, получаем

$$t(x) = \frac{x}{v_1 sin\alpha} + \frac{L-x}{v_2 sin\beta}$$

По условию задачи - это минимальное время. Известно, в точках максимума или минимума первая производная непрерывной функции должна быть равна нулю, то есть, *t'(x) = 0*. Условие минимума времени:

$$t'(x) = \left(\frac{x}{v_1 sin\alpha}\right)' + \left(\frac{L-x}{S_2 sin\beta}\right)' = 0$$

$$\frac{1}{v_1 sin\alpha} + \frac{-1}{v_2 sin\beta} = 0$$

$$\frac{1}{v_1 sin\alpha} = \frac{1}{v_2 sin\beta}$$

Отсюда получаем соотношение: $\frac{v_2}{v_1} = \frac{sin\alpha}{sin\beta}$

Это сильно напоминает закон преломления света, где n – показатель преломления света на границе двух сред:

$$n = \frac{v_2}{v_1} = \frac{sin\alpha}{sin\beta}$$

Чтобы определить точку пересечения границы курьером, нужно выразить и вычислить значение x из уравнения, в котором t имеет минимальное значение. Для этого попробуем вычислить время геометрически. Выразим полное время через сумму $t = t_1 + t_2$, где

$$t_1 = \frac{\sqrt{h_1^2 + x^2}}{v_1}; \tag{1}$$

$$t_2 = \frac{\sqrt{h_2^2 + (L-x)^2}}{v_2}; \tag{2}$$

$$t = t_1 + t_2; \tag{3}$$

$$t = \frac{\sqrt{h_1^2 + x^2}}{v_1} + \frac{\sqrt{h_2^2 + (L-x)^2}}{v_2} \tag{4}$$

Возьмем первую производную

$$t' = \left(\frac{\sqrt{h_1^2 + x^2}}{v_1}\right)' + \left(\frac{\sqrt{h_2^2 + (L-x)^2}}{v_2}\right)' = 0$$

$$\frac{2x}{2v_1\sqrt{h_1^2 + x^2}} + \frac{-2(L-x)}{2v_2\sqrt{h_2^2 + (L-x)^2}} = 0$$

Окончательно получаем:

$$\frac{x}{v_1\sqrt{h_1^2+x^2}}+\frac{x}{v_2\sqrt{h_2^2+(L-x)^2}}=\frac{L}{v_2\sqrt{h_2^2+(L-x)^2}} \qquad (5)$$

После избавления от радикалов в этом уравнении неизвестная величина x имеет степени: x^4, x^3, x^2, x, следовательно, значение x необходимо подбирать экспериментально.

Как видим, аналитически решить задачу невозможно, для этого нужно провести многократные вычисления, пересекая границу OL в разных точках. Задача решается методом подбора значений переменной x, удовлетворяющих условию задачи. В итоге можно заполнить таблицу и в таблице найти строку с минимальным значением затраченного времени. При этом точного значения x найти невозможно. Поэтому допускается неизбежная погрешность ±Δx, которая зависит от количества шагов. Таким образом, данная задача решается только с помощью компьютерной модели. Исполнительный алгоритм, представленный на рисунке (рис. 5), полностью копирует шаги, сделанные выше (рядом с формулами поставлены порядковые номера и в этом же порядке они включены в программу компьютерной модели.

$x = 0, \quad \Delta x = 50$

$$t_{min} = \frac{h_1}{v_1} + \frac{\sqrt{h_2^2+L^2}}{v_2}$$

1) $t_1 = \frac{1}{v_1}\sqrt{h_1^2 + x^2}$

2) $t_2 = \frac{1}{v_2}\sqrt{h_2^2 + (L-x)^2}$

3) $t = t_1 + t_2$

Если $t < t_{min}$, то $x_c = x$, $t_{min} = t$

$x = x + \Delta x$

Вывод: $x_c =$ $\quad t_{min} =$

Рис. 5. Исполнительный алгоритм задачи «Курьер»

Причем, эти формулы используются многократно (в цикле).

1) $t_1 = \frac{\sqrt{h_1^2+x^2}}{v_1}$ $\qquad$ 2) $t_2 = \frac{\sqrt{h_2^2+(L-x)^2}}{v_2}$ $\qquad$ 3) $t = t_1 + t_2$

Эти три команды позволяют вычислить затраченное время при пересечении курьером границы в любой точке x. На каждом шагу результат t сравнивается с предыдущим результатом и, если величина t оказалась меньше предыдущего, за минимальное значение принимается текущее значение $t_{min} = t$. Но, если следующее значение t окажется больше t_{min}, значение t_{min} не меняется и сохраняется до конца выполнения программы.

В приведенной таблице (рис. 6.) отражены результаты 10 шагов и из таблицы видно, что минимальное значение t =263,6 с соответствует пересечению курьером границы в точке x = 850 м. Точность результата можно повысить. Для того, чтобы найти x с точностью $\Delta x = \pm 1$ м, нужно провести 1000 вычислений с шагом $\Delta x = 1$ м. Уменьшая шаг поиска до сантиметров, миллиметров и так далее, можно достигнуть высокой точности результата.

```
    - - - - - - - - -
    L. = 1000: H1 = 300: H2 = 200
    V1 = 5: V2 = 3 X = 0
    T1 = H1 / V1 + SQR (L ^ 2 + H2 ^ 2) / V2
    T2 = SQR (H1 ^ 2 + L ^ 2) / V1 + H2 / V2
    TMIN = T1
    PRINT "    "; T1, T2, TMIN: PRINT: PRINT
    PRINT "    "; "X = "; X; "  ", "T = "; TMIN
    X = 0: DX = 50
    100: X = X + DX
    IF X > 1000 THEN 200
(1) T1 = SQR (H1 ^ 2 + X ^ 2) / V1
(2) T2 = SQR ((L - X) ^ 2 + H2 ^ 2) / V2
(3) T = T1 + T2
    PRINT "    "; "X = "; X; "  ", "T = "; T
    'IF T > TMIN THEN GOTO 200
    IF T < TMIN THEN TMIN = T: XC = X
    GOTO 100
    200: PRINT "XC = "; XC, "TMIN = "; TMIN
    END
```

X	=	50	T	=	384.4358
X	=	100	T	=	370.5637
X	=	150	T	=	358.1529
X	=	200	T	=	346.9847
X	=	250	T	=	336.8387
X	=	300	T	=	327.5231
X	=	350	T	=	318.8866
X	=	400	T	=	310.8185
X	=	450	T	=	303.2449
X	=	500	T	=	296.1245
X	=	550	T	=	289.4473
X	=	600	T	=	283.2353
X	=	650	T	=	277.5492
X	=	700	T	=	272.5005
X	=	750	T	=	268.2737
X	=	800	T	=	265.161
X	=	850	T	=	263.6109
X	=	900	T	=	264.2723
X	=	950	T	=	267.967
X	=	1000	T	=	275.4728

Рис. 6. Компьютерная модель задачи «Курьер»

Пока окончательного варианта оформления или представления исполнительного алгоритма пока не существует, но такая необходимость имеется. Мы считаем, что он может быть невидимым, но легко узнаваемым в решениях большинства задач из задачника, Но в случаях сложных и трудных задач для создания компьютерной модели для проведения экспериментов, его наличие просто необходимо, и, только в этом случае, наличие исполнительного алгоритма обязательно.

Тем не менее, когда это будет принято большинством учителей и преподавателей, можно будет досконально описывать свойства исполнительного алгоритма. Исполнительный алгоритм разрабатывается для решения конкретной задачи и для других задач непригоден, он указывает на строгую последовательность действий компьютера при ее решении.

Предлагаемый нами, метод разработки и применения исполнительного алгоритма компьютерной модели вносит много положительного в процесс решения физических задач с помощью компьютерных технологий.

«Плюсы» исполнительного алгоритма компьютерной модели:

• Исполнительный алгоритм состоит только из узнаваемых формул основных физических законов (сразу видно используемый закон, принцип).

• Высокая наглядность: видны все нюансы и особенности работы компьютерной модели.

• Пригоден для любого вида моделирования (численного, табличного, имитационного).

• Понятен для учащихся, так как используется запись формул и уравнений на привычном математическом языке.

• Исполнительный алгоритм пригоден для любого программного обеспечения на разных языках программирования (Pascal, Qbsaic, VisualBasic, Phyton, Excel и др.).

• Чтобы создать исполнительный алгоритм не обязательно знать основы программирования.

«Минусы» исполнительного алгоритма компьютерной модели:

• Исполнительный алгоритм для каждой задачи создается отдельно, индивидуально. Не существует универсального исполнительного алгоритма. Его нельзя копировать с одной задачи на другую.

• Обязателен при решении сложных задач с многочисленными вычислениями и решения задач на основе дифференциальных и интегральных уравнений.

• Несколько вариантов, исполнительный алгоритм зависит от создателя и выбранного способа решения задачи. Дело в том, что одну и ту же задачу можно решить разными способами. Прежде, чем разрабатывать исполнительный алгоритм, предварительно нужно решить задачу или выбрать соответствующий метод решения.

Мало времени уделяется для решения задач, поэтому наш метод позволит глубже рассмотреть физические закономерности, научить разбираться в них и применять их к анализу физических явлений, к практическим вопросам.

Выводы. В заключение следует отметить, что наличие исполнительного алгоритма ускоряет процесс разработки компьютерной модели решения задачи. Кроме того, разработка алгоритма формирует у учащихся умение организовать физический эксперимент, умение руководить, с целью достижения поставленной цели. Верим, что идея исполнительного алгоритма для компьютерного моделирования решения задач найдет положительный отклик среди учителей физики, математики и информатики и получит достойное продолжение.

Литература

1. Бурсиан Э. В. Задачи по физике для компьютера: Учеб. пособие для студентов физ.-мат. фак. пед. ин-тов. – М.: Просвещение, 1991. – 256 с.
2. Гутман В. И., Мощанский В. Н. Алгоритмы решения задач по механике в средней школе: Кн. для учителя. - М: Просвещение, 1988. - 95 с.
3. Каменецкий С. Е., Орехов. В. П. Методика решения задач по физике в средней школе. - М.: Просвещение, 1987. - 336с.
4. Майер Р. В. Решение физических задач в электронных таблицах Excel: учебное пособие [Электронное учебное издание на компакт-диске]. – Глазов: Глазовск. гос. пед. ин-т, 2016. – 14,0 Мб.
5. Тулькибаева Н. Н., Фридман Л. М., Драпкин М. А., Валович Е. С., Г. Д. Бухарова Решение задач по физике. Психолого-методический аспект. Челябинск: Изд. ИГПИ «Факел», ЧВВАИУ и Урал. Гос. проф. пед. ун-та, 1995. - 120с.