NEUROSCIENCE EXPERIMENTS FOR MATHEMATICS EDUCATION

FRANK QUINN

Abstract. These experiments are explicit, designed to give clear outcomes, and have significant impact on educational practice.

The first group explores difficulties caused by mixing cognitively distinct activities. Topics are multiplication of polynomials, word problems, and customary notation and usage. The second group explores subliminal (invisible or unanticipated) learning. Topics are algebra in elementary arithmetic, learning multiplication facts, and kinetic reenforcement of function graphs.

For quite some time it has seemed that cognitive neuroscience should contribute powerfully to math education, but this has not happened. This article shows that a connection is possible. A companion article (*Mathematics education versus cognitive neuroscience*) explores reasons for the failure to date; there seem to be deeply rooted structural problems.

Contents

1. Outline

These experiments contrast standard educational approaches that seem to cause problems, with alternatives adapted to mathematical and cognitive structure. Many of the alternatives have been used successfully in individual cases, so the job of neuroscience is not so much to detect the effects as to quantify them and clarify

Date: Draft, February 2011.

the mechanisms. Moreover, the tools of neuroscience are difficult and indirect so specific tasks are designed to maximize expectation of clear outcomes.

The specific examples might be thought of as opening moves in collaborations, addressed to neuroscientists. They are sensible of the needs and limitations of neuroscience, and informed by neuroscience studies, but for the most part are not specific about neuroscience protocols because this is the job of potential collaborators. Cognitive, mathematical, and educational issues are discussed in some detail because these are responsibilities of the mathematical-educational partner.

These studies should have significant implications for educational practice. A companion essay [16](c) explores why educationally-oriented neuroscience studies to date have had so little impact. In this context, a motivation for developing high-impact examples was to test this picture, and in particular see if there are barriers not yet identified.

1.1. **Background.** A few clarifications about background and constraints for these proposals.

1.1.1. *Diagnostic experience.* These proposals draw on extensive diagnostic work with students. The format is a session in which a student describes work in which something has gone wrong, while the helper listens and diagnoses the source of the problem. The helper then (briefly) explains the error and how to avoid it in the future. This gives a much finer-grained view of learning and its problems than does traditional teaching.

A consequence is that the concerns here are only indirectly related to teaching, and are often inconsistent with mainstream (teaching-oriented) ideas about learning.

1.1.2. *External working memory.* Mathematics uses written scratch work as external working memory, and many neural processes are adapted to this.

Suppose, for example, that one step in a procedure collects numbers to be added, and the next step is to add them. The first step is purely symbolic, unconcerned with specific number properties. The second step requires processing them as numbers, but little or no engagement of organizational or symbol-manipulation facilities. The two tasks use different neural processes. Written work is used for reformatting (different processes read symbols in different, task-appropriate ways) and inter-process communication. This is much more efficient and accurate than trying to do it internally.

Consequences for neuroscience studies are:

- Subjects must be able to do scratch work during all procedures.
- A time-stamped record of external work provides a window on internal activity.
- One goal is to understand the interactions between neural activity and external memory. Effectiveness can often be substantially improved by optimizing procedures and notation for external-memory use.

1.1.3. *Mathematical structure.* This is a cautionary note. Mathematical work is both enabled and constrained by mathematical structure, and experimental design and interpretation must be carefully adapted to this structure. The discussion here only hints at the constraints involved, and these should be fully understood before modifications or variations are undertaken.

1.2. **Cognitive interference, outline.** The four examples in §2 concern interference between subtasks of a task. In a nutshell, mixing or switching between subtasks can reduce success and limit scope of application. In some cases procedures or algorithms could be reorganized to separate or eliminate such tasks, but this is constrained by mathematical structure and the need to avoid introducing new cognitive difficulties. The challenge is to identify interference severe enough to need change, in circumstances where structure permits it.

1.2.1. *Polynomial multiplication.* The example in §2.1 concerns conflict between the organization and calculation aspects of multiplication. The experiment uses multiplication of polynomials by high-school or beginning college students, comparing performance with standard (mixed-task) methods and a task-separated algorithm. The immediate outcome should be better methodology for polynomials. A long-term goal is better methodology for multi-digit multiplication in elementary mathematics, and this is taken up in §3.1. The polynomial version provides a simpler (in some cognitive senses) and slower (for imaging purposes) model.

The context for the study is task switching between two basic task types. There is an extensive literature on mechanisms and costs of switching in very simple tasks, see the review [12]. The ACT–R computer model [1], [2] has been extensively tested seems to model some elementary tasks reasonably well, c.f. [20]. The work done so far is only marginally relevant, however. It corresponds, roughly speaking, to discrete behavior of matter at very small scales, while we are concerned with statistical behavior at significantly larger scales.

A useful general conclusion from cognitive psychology [14] and clearly visible in small-task studies, is that our thinking is essentially single-track. Ample working memory and frequent switching between superficial tasks give the impression of "multitasking", but this is mostly an illusion [24][c]. Specifically, if we switch from task A to a different task B but know we will shortly be doing A again, we usually cannot economize by keeping task-A instructions loaded but off-line. Instead we have to flush task-A material, load B, and when B is complete, reverse the procedure. Further

- "Flushing" task A may involve *inhibiting* task-A instructions, not just emptying a buffer. Some errors (e.g. adding instead of multiplying) result from incomplete inhibition.
- Residual effects of this inhibition can slow or complicate reloading for the next A task. In other words, repeated switching reduces the effectiveness of working memory [11]
- Following an A task by another instance of A usually requires less reorganization and has lower costs.

Algorithms with subtasks $ABABAB\dots$ usually cannot be reorganized as $AAABBB$ for mathematical reasons, but when it is possible it should have cognitive benefits. The multiplication proposal is of this type.

1.2.2. *Interference from customary usage.* Customary usage often interferes cognitively with mathematical work. This is usually easy to fix by changing notation, avoiding customary forms, or introducing translation as an explicit separate step. But customary usage is, essentially by definition, transparent to adults. As a result the problems caused by it are invisible, and attempts to deviate from customary usage make adults uncomfortable and are resisted. The role of neuroscience is to help locate these problems, and unequivocally identify them as problems.

For example $\sqrt{A}$ is a customary notation for $A^{1/2}$. The exponential form fits into general patterns and is usually easier to manipulate. The customary notation alerts us to the possibility of using special properties of square roots. The best practice in this case seems to be to use the customary notation so we get the special-property alert, but teach students that in most problem types the first step should be to rewrite it in exponential form. In contrast the customary notation $\sqrt[3]{A}$ for cube roots does not have benefits that compensate for translation overhead. This should always be written as $A^{1/3}$.

§2.3 concerns rather severe problems with customary use of parentheses. The problem is discussed because it is important and effects several other proposals. No specific experiment is proposed, however, because it has been difficult to find one with clear and useful outcomes.

§2.4 concerns the translation overhead of irregular customary names for integers (e.g. 'thirteen' for 13). Cross-cultural and imaging studies suggest that short-term working memory is mostly verbal, even when working with numbers. For instance, the total length of names for things is often a stronger limit than the number of things. Another clue comes from the additional difficulty children have in learning to count in languages with irregular customary number names.

A conclusion is that counting and arithmetic might—in some languages at least—be simplified by the use of "math names" for numbers. The experiment explores this through its effect on iterated addition.

1.2.3. *Word problems.* The example in §2.2 concerns cognitive interference between the modeling and analysis aspects of word problems. The experiment compares performance and neural activity in the standard (mixed-task) approach and a task-separated modeling approach. The short-term goal is to show that the educational approach is counterproductive. The longer-term goal is to explore ways to use word problems effectively in elementary education.

Educators see word problems as essentially mathematical; a different format rather than a different activity. As a result educators encourage a gestalt approach in which students "develop strategies" to work directly with the formulation of the problem. Students find this hard, and accessible problems have either mathematical or modeling component (or both) trivial.

Mathematicians and professional users of mathematics divide real-world applications into two steps: 'modeling' translates physical data to a self-contained symbolic formulation called 'the model', and then the model is analyzed mathematically. These two steps use very different methods and, technically, the modeling step is not mathematical. Diagnostic experience with students suggests that modeling and analysis are also quite different cognitively. Mixing seems to cause interference considerably stronger than that seen in multiplication, and success in science, engineering, and related mathematics, requires use of task-separated modeling.

1.3. **Subliminal learning and reenforcement, outline.** We are concerned with learning that takes place during an educational activity, but that is invisible to the student, and frequently to educators as well. There are two variations: subliminal learning from the content; and learning that depends in an unrecognized way on the human interaction aspect (e.g. kinetic or verbal) of the activity.

When mathematics is done by hand, a lot of activity comes as packages that are activated by simple goals. New methods—especially technology—cause these packages to come apart, and important subliminal learning may be lost. For instance "find 365×86" requires a lot of neural activity when done by hand and rather less when a calculator is used. Is the extra activity pointless, or are there important benefits of the package rather than the number obtained?

Diagnostic work with students suggests that there is quite a lot of subliminal learning in by-hand elementary mathematics that is lost in calculator curricula [16](d). A goal of the experiments is to fix this: understand instances well enough to design programs that use technology and also provide this learning. Curiously, this should also enable improvement of traditional programs. Subliminal learning in by-hand work is usually accidental and inefficient. Better understanding should enable more efficient approaches, either with or without technology.

The role of neuroscience is this: neural effort in well-learned skills is usually focused in a small number of regions. Early attempts usually recruit much wider activity. Development requires exercising the necessary regions and connections between them, and also requires suppression of recruitment of unneeded regions. Neural activity alone is not a definitive guide to learning, but it gives excellent clues:

- Activities that exercise appropriate regions probably contribute to skill development.
- Activities that do not engage these regions cannot contribute much to skill development.
- Too much emphasis on activities that consistently engage unnecessary regions may impede skill development.

1.3.1. *Subliminal algebra in integer arithmetic.* This experiment concerns subliminal internalization of algebraic structure from by-hand integer multiplication. The point is that the symbols we write to represent numbers are symbols, not numbers, and by-hand arithmetic involves a lot of symbol manipulation. Students seem to internalize some of the algebraic structure used in these manipulations.

The place-value notation presents integers as polynomials in powers of ten, with single-digit coefficients. For instance $438 = 4 \cdot x^2 + 3 \cdot x^1 + 8 \cdot x^0$, with $x = 10$. The standard algorithms for multi-digit multiplication essentially multiply the corresponding polynomials and then evaluate at 10.

The experiment in 3.1 has two parts. The first compares neural activity in 3×3-digit multiplication by hand, and with a calculator. An objective is to see to what extent the hand work recruits neural regions used in algebra, and more specifically in polynomial multiplication.

The second part explores the use of a task-separated algorithm modeled on the polynomial algorithm of §2.1. The first version is for hand use. It requires more writing than the traditional algorithm but should display the structure more clearly and be easier to use accurately. The second version uses a calculator, but in a way that still requires expansion and display of algebraic structure. The objectives are to assess potential cognitive benefits by comparing neural activity with that associated to standard by-hand multiplication.

There are many studies of numerical multiplication, c.f. [9, 20]. Unfortunately conceptual and methodological weaknesses [21], [16](c) render these only marginally relevant.

1.3.2. *Subliminal learning of number facts.* A certain amount of transparent mental arithmetic is vital for learning in algebra and beyond, as I explain below. A consequence is that calculators cannot substitute for automatic recall of single-digit multiplication facts. But this does not mean we are stuck with rote memorization. The proposal in §3.2 section explores a subliminal approach using the algorithm described in §3.1.

The reason mental arithmetic is necessary is that examples illustrating mathematical procedures almost always have arithmetic subtasks. In practice we can contrive these subtasks to be fairly simple; see for instance the example used to illustrate the task-separated polynomial multiplication algorithm in §2.1.2. Algorithms are usually learned correctly as long as the arithmetic is complex enough to avoid misleading numerical coincidences. After the algorithm is internalized it can be used for arithmetically complex problems, even if coefficient calculations require calculators or extensive scratch work but for initial learning, transparent mental arithmetic is essential.

The amount of mental arithmetic needed is a compromise between what students can learn relatively easily, and how simple the examples can be contrived to be and still effectively illustrate structures. The standard compromise is that the following should be fully transparent:

(1) Addition and subtraction of single-digit integers.
(2) Addition of four or five single-digit integers, or a three-digit and a two-digit integer[1].
(3) Multiplication of a single-digit and a two-digit integer.

In addition, multiplying a one-digit and a three-digit integer should not be a distraction, though it need not be fully transparent. Multiplying two 2-digit integers is cognitively more complex (see §2.1.1) but should still not disrupt the train of thought. We can almost always contrive to avoid larger multiplications. Students should certainly be able to do them, but transparency is not needed.

Finally, the *formal structure* of arithmetic should be automatic enough that a few symbols will not cause problems.

1.3.3. *Kinetic reenforcement in graphing.* This experiment concerns reenforcement of internalization of geometric structure of function graphs, by the kinetic aspect of by-hand graphing. In non-technology programs, both assignments and testing require drawing by hand. In programs using technology, student work has visual outcomes, and testing is also usually visual (choose the correct graph among a number of alternatives).

Diagnostic experience suggests that many graphing-calculator trained students cannot either verbally describe or qualitatively sketch standard curves. When they do try to draw pictures they often reproduce a calculator display, to scale, with typical poor choices of range and microscopic features of interest. In other words they have not internalized the qualitative geometric structure. It seems that the kinetic aspect of drawing powerfully (and subliminally) reenforces learning of qualitative structure, and some students seem unable to learn without this reenforcement.

A general context is that serious learning benefits from, and often requires, active reenforcement. Recent studies ([18], [19]) report that young children do not learn

[1] A better algorithm would probably put larger addition problems within easy reach, see §2.1.5, but this does not seem to be a bottleneck in actual use.

from videos. To learn vocabulary, for instance, they must say the word, not just hear it. Verbal reenforcement seems to be more effective when 'social cognition' facilities are engaged by the presence of an attentive human, and this may be the primary mechanism in some cases. None of this should be a surprise. Children in rural areas learn their local dialects but usually not (in the US) standard English, even though they hear as much or more standard English on television. Similarly, what a child sees makes far less impression that what he draws or writes himself.

The experiment uses two versions of a brief lesson on qualitative graphs of sums of functions. The first version uses a typical visual computer-graphic approach, and the second a hand-drawing approach. Students are quizzed using version-appropriate methods: visual multiple-choice in the first case, drawing in the second. Finally they are tested with the opposite methodology.

The questions concern similarity and differences in neural activity in the two modes, and transfer of learning from one mode to the other. Diagnostic experience suggests that kinetically-reenforced learning should transfer, visual learning usually will not. This experiment is more complex than the others because the questions concern neural activity during learning, not just during use of a learned procedure.

This is the end of the outline.

2. Cognitive interference

Mixing different tasks often slows and degrades performance in both. It seems likely that such interference has a neural basis. Understanding this should enable design of algorithms and procedures better adapted to humans use, mainly by separating internal tasks and using scratch work for reformatting and high-precision interprocess communication. The proposals address two instances in which interference has been observed: multiplication and word problems. See §1.2 above for an outline.

2.1. **Cognitive interference in multiplication.**

There are two important cases that use essentially the same algorithm: multi-digit integer multiplication in elementary school, and polynomial multiplication in high school and college. We begin with polynomials because:

- the polynomial version is actually a bit simpler because there are no overflow problems associated with converting polynomial-like outcomes into place-value integer notation;
- the separated polynomial tasks take long enough to be imaged by fMRI, and this is unlikely with integer multiplication;
- more-extensive scratch work (external working memory) can be used to correlate cognitive and neural activity;
- high-school or college students are more consistent and cooperative experimental subjects;
- arithmetic skills of older students are already well-established and stable, and should produce clearer and more consistent signals.

Another reason to begin with polynomials is that the problem seems to involve a genuine limit on human ability: experienced professors of mathematics seem to have as much trouble as students with the mixed-task algorithm, and get as much benefit from the task-separated version. This should mean that the underlying neural issues should be relatively uniform and clear. In contrast, the interference experienced by children with multi-digit integer multiplication can be eventually be managed, so may have a developmental component. In fact there are likely to be a number of different difficulties, and before we can assess any of them we must understand the mature endpoint. Further, the proper course of action may be unclear. If the problem is *only* developmental then finding ways to speed development would probably be more useful than tinkering with algorithms.

The integer case is discussed further in §3.1. Detailed mapping of component functionality is discussed in [16](e).

2.1.1. *Sample problems.* These examples show escalating conflict between organization of the polynomial structure, and coefficient arithmetic. Coefficients are contrived so individual operations are easy; difficulties come from mixing rather than from individual operations.

1) Write $(3x^2)\,((2-a)x^3)$ as a polynomial in x.

Note that "simple arithmetic" in coefficients may include symbols, to emphasize that we need transparent internalization of structure (associative, distributive etc.), not just number facts. This example has one coefficient operation and one polynomial operation: they are perforce separated and there is little conflict.

2) Write $(3x^2 - x + 5a)\,((2-a)x^3)$ as a polynomial in x.

The result has three terms. The standard practice is to do coefficient arithmetic as each term is generated, so there are two arithmetic interruptions of the polynomial procedure. There is relatively little interference, partly because there are few interruptions. Another reason is that the structure of first term provides a template for sequential organization of the task. Minor interference is suggested by more-frequent sign mistakes with the -1 coefficient on x in the first term, as compared to errors in isolated arithmetic tasks.

3) Write $(3x^2 - x + 5a)(x^3 + (2-a)x^2 - a)$ as a polynomial in x.

Simple expansion gives nine terms, with eight interruptions for coefficient arithmetic. Moreover the data is a 3×3 array so a strategy for organization as a sequential task must be devised. Finally, terms with the same coefficient have to be collected and combined. The success rate is low and errors in both organization (missed terms) and arithmetic are common. The difficulty comes from the algorithm rather than the problem itself, however, as we see next.

2.1.2. *Task-separated algorithm.* The basic plan is to separate different tasks as completely as possible. In polynomial multiplication, organizational work related to the polynomial structure should be completely separated from coefficient arithmetic, and multiplication and addition separated in the arithmetic. This is illustrated with problem (3) above: write

$$(3x^2 - x + 5a)(x^3 + (2-a)x^2 - a)$$

as a polynomial in x.

Step 1: A preliminary scan shows that the output will be a polynomial of degree 5. Set up a template for this:

$$x^5(\qquad) + x^4(\qquad) + x^3(\qquad) + x^2(\qquad) + x^1(\qquad) + x^0(\qquad)$$

Step 2: Fill in the blanks one at a time. For example, the terms with total exponent 3 are obtained as follows: the highest-order term in the first factor is x^2; record its coefficient (3). The complementary exponent is 1, but there is no x^1 term in the second factor so we record 0. Move to the next lower power in the first factor and the next higher in the second, and record coefficients $(-1)(2-a)$. Continue to get $((3)(0)+(-1)(2-a)+(5a)(1)$. Put everything in parentheses, and not do *any* arithmetic on the fly. Do not, for example, omit the 3 coefficient on x^2 because there is no complementary term in the second factor, and do not write $(5a)(1)$ as $5a$.

- This enables reading the coefficients only as strings to be copied, with no arithmetic significance. This reduces cognitive overhead.
- Even completely trivial arithmetic requires a momentary change of gears, and watching for an opportunity to do it is a distraction.

The outcome is:

$$x^5((3)(1)) + x^4((3)(2-a) + (-1)(1)) + x^3((3)(0) + (-1)(2-a) + (5a)(1))+$$
$$x^2((3)(-a) + (5a)(1)) + x^1((-1)(-a)) + x^0((5)(-1))$$

Step 3: Do multiplications:

$$x^5(\underbrace{(3)(1)}_{3}) + x^4(\underbrace{(3)(2-a)}_{6-3a} + \underbrace{(-1)(1)}_{-1}) + x^3(\underbrace{(3)(0)}_{0} + \underbrace{(-1)(2-a)}_{-2+a} + \underbrace{(5a)(1)}_{5a}) + \dots$$

The process and notation is designed to avoid organizational activity: input for each operation is in standard position in the visual field, the underbrace specifies the input so it does not have to be reconstructed for other steps or checking, and output is put in a standard place.

Step 4: Do additions.

$$x^5(\underbrace{(3)(1)}_{3}) + x^4\underbrace{(\underbrace{(3)(2-a)}_{6-3a} + \underbrace{(-1)(1)}_{-1})}_{5-3a} + x^3\underbrace{(\underbrace{(3)(0)}_{0} + \underbrace{(-1)(2-a)}_{-2+a} + \underbrace{(5a)(1)}_{5a})}_{-2+6a} + \dots$$

Again the process and notation minimize organizational activity that would interfere with arithmetic.

2.1.3. *Experiment.* The subjects should be high-school students who have been successful in a standard algebra curriculum, or students in first-year college calculus (i.e. not remedial). Proficiency with problems like (2) above might be used as a criterion. They should also be screened for dependence on calculators for basic arithmetic (see below). They are asked to work problems similar to the one above, using standard methods. They are then taught the task-separated version, and after enough practice to become familiar with it, they are imaged working similar problems with this methodology.

- To keep the picture clear the arithmetic should be kept minimal. Multiplication of multi-digit integers, for instance, would produce a small version of the entire process.
- Half the problems should have numerical coefficients, half have symbols in the coefficients (as in the example).
- Subjects should be told that accuracy is more important than speed. Errors due to speed or carelessness will mask significant features of more intrinsic mistakes [17, 7, 6].

- Scratch work should be videotaped and time-stamped, for correlation with imaging results.

fMRI should provide general information about the areas used and the degree of usage, c.f. [9, 20]. It would be useful if there is an easily-identified MEG or EEG signature of major task switching—a mental shifting of gears—in the task-separated versions, c.f. [22].

2.1.4. *Calculator version.* The core experiment concerns students with good manual arithmetic skills. If resources permit, it can be expanded to include students with similar proficiency but who use calculators for numerical work. Let them use calculators as they like during the trials, and record this use. Expected differences are described below.

2.1.5. *Analysis.* The basic plan is to look for neural and performance differences between the standard and task-separated versions. The expectation (based on diagnostic work with students) is that performance should be significantly better with the task-separated version, and the hypothesized reason is that the task-separated version reduces interference caused by arithmetic interruptions of the polynomial organizational task. A qualitative picture should emerge reasonably quickly. It might be possible to directly explore interruptions and their short-term neural consequences by carefully correlating imaging with scratch work.

The above is the basic plan. We now discuss potential complications and refinements.

First, there may be a sub-population with substantially better performance with the modified algorithm. The goals are algorithms that benefit everyone when used as the standard approach, but it is unlikely that everyone will benefit when they are used as a retrofit. The cognitive interference signal should be clearest in the high-performance group. Note that subjects cannot be screened in advance for quick adaptation because the control experiment (using standard techniques) becomes impossible after the modified algorithm is taught. Predictors of success found after the fact, however, would certainly be useful.

Calculator arithmetic requires a significant attention shift and input/output processing, and there are a great many discrete arithmetic tasks in these problems. It is hard to imagine that calculator use could become so transparent that this would not be a source of interference. The prediction, therefore, is that students who actually make substantial use of calculators during the trials will have lower success with any form of these problems.

Next, if at all possible, individual variation in the task-separated version should be investigated. Currently the statistical techniques used to analyze data have a built-in assumption that everyone does these things in essentially the same way. Variation is treated as noise. The data showing that multiplication facts are stored in the angular gyrus using verbal memory, for instance, demonstrates that this is the dominant mode. But is it really true that no-one uses visual memory for this? Understanding variation in successful learning is essential for understanding all the barriers to success, and the separated tasks may be long and uniform enough to permit this. Again, students who use calculators are likely to have significantly different characteristics.

The number and nature of mistakes made is more significant than time required to complete the tasks. Time measurements might be useful for comparing different task instances done by one individual, however.

Finally, it will be very important to assess the effects of symbols in the coefficients. The hypothesis suggested by behavioral data is that students who have effectively internalized the symbolic structure of arithmetic should show little difference in either performance or neural activity. There is some support for this in very simple tasks [2], [23]. Conversely, students who have not internalized this structure, or who think of symbols and numbers as essentially different, will find symbolic coefficients significantly more difficult. Most calculator users are likely to be in this group.

2.2. **Cognitive interference in word problems.** The modeling and analysis components of word problems seem to interfere when mixed, and this interference is often very strong. This is explored through comparison of student work using standard (mixed-task) and modeling (task-separated) procedures. See §1.2.3 for discussion.

2.2.1. *Sample problem.* The following have the same mathematical core.

Food version: A basket contains six loaves of bread. Half of these are put in another basket that already contains nine loaves. Then one-third of the total contents of the second basket is put in the first. How much bread ends up in the first basket?

Social version: Jen and Brad have six loaves of bread. Brad takes half with him when he leaves to share *everything* with Angelia, who already has nine loaves. Jen's lawsuit against Brad and Angelia is settled by giving her one-third of Brad and Angelia's bread. How much bread does Jen end up with?

Money version: A basket contains six dollars. Half of these are put in another basket that already contains nine dollars. Then one-third of the total contents of the second basket is put in the first. How much money ends up in the first basket?

These are easy to model and solve, but difficult with the gestalt approach because interpretation and calculation are mixed.

2.2.2. *Task-separated (modeling) version.* Let A denote the bread in the first basket, with subscripts $1, 2, 3$ corresponding to the three times. B_i similarly denotes the bread in the second basket. Translating the data for the beginning state gives:

$$A_0 = 6,\ B_0 = 9.$$

Changes that give the second state translate as:

$$A_1 = A_0 - \frac{1}{2}A_0,\ B_1 = B_0 + \frac{1}{2}A_0.$$

Finally changes that give the third state give:

$$A_2 = A_1 + \frac{1}{3}B_1,\ B_2 = B_1 - \frac{1}{3}B_1.$$

This is a symbolic form (model) suitable for mathematical analysis. After doing a few of these they become immediately recognizable as short recurrence relations.

Analysis proceeds in two stages; first substitute in two steps to reduce to a numerical problem:

$$A_2 = A_1 + \frac{1}{3}B_1 = (A_0 - \frac{1}{2}A_0) + \frac{1}{3}(B_0 + \frac{1}{2}A_0) = 6 - \frac{1}{2}(6) + \frac{1}{3}(9 + \frac{1}{2}(6))$$

and finally do the the arithmetic. See §2.2.5 for discussion of cognitive and conceptual features.

2.2.3. *Experiment.* The subjects are high-school students who have been successful in a standard algebra curriculum. They are imaged working word problems with the standard reasoning-in-context approach. They are then taught the task-separated version, and after enough practice to become familiar with it, they are imaged working similar problems with this methodology. They should be asked to give the model as part of the solution (to ensure actual separation), and some problems should ask only for the model.

In both trials, problems to be worked should be interspersed with controls in which students are asked only to identify problem type (food, social, etc.).

Finally, subjects should be interviewed before and after the imaging trials. Pre-trial questions would concern attitudes toward word problems (enjoy, dread, etc.), neutrally probe reasons (actually interesting, easy grades because the math is trivial, believe teachers' assertion that they are important, etc.), and ask the subject's impression of his general competence and success rate. Post-trial questions would include feelings about task separation (helps, is a waste of time), and assess changes in interest and feelings of competence.

There are two points to the interviews. First, is there a correlation between reduced cognitive interference and increased interest or confidence? Second, assertions about motivation and "real-life" relevance are used to justify current use of word problems; there is essentially no mathematical justification because the analytic aspects are trivial. It is therefore important to honestly assess student motivation and feelings of "relevance", and determine how a different approach might effect them.

2.2.4. *Analysis.* Unseparated work should show extensive activity, probably including prefrontal recruitment to sort out confusion from interference. Active areas will probably depend on the nature of the problem, and different types should be analyzed separately to see this. The social version, for instance, should engage neural structures devoted to interaction with others of our species. Comparison with type-identification versions should reveal activity specific to the mathematical task. Questions:

- Do some types interfere with mathematical activity more strongly than others (i.e. have lower success rates)?
- Do different types lead to differences in the mathematical components, as revealed by subtracting type-identification responses?
- Is there systematic variation, for instance sex differences in responses to social versions, or socioeconomic level effects in responses to food or money versions? If so, how do these correlate with success rates?

Subtasks in unseparated work will have irregular timing and sequencing, and will be hard to image. This is not a big problem.

Task-separated versions should show clearly-defined shifts between modeling and analysis. Questions are:

- How do the neural areas and degrees of activation compare to the non-separated versions? For instance, are the same areas used, just in sequence rather than simultaneously?
- Modeling has some symbolic activity, and this should be revealed by subtracting type-identification responses. Where does this take place, and is it essentially the same for all problem types?
- The symbolic aspect of modeling seems not to interfere with other parts of the process, as long as no analysis is done. Is this true on the neural level, or does it reveal interference too mild to be obvious?

2.2.5. *Further discussion.* The immediate cognitive benefit of the task-separated version is that translation and analysis are both routine and reliable, and can be extended. Adding another layer, for instance if Brad goes back to Jen and there is another redistribution of bread, could easily be done in the task-separated version but would be a serious challenge with the gestalt approach.

Modeling also has conceptual benefits. The model displays the mathematical structure as a recurrence relation rather than a sequence of arithmetic operations. Similar models describe superficially different problems, showing the underlying unity and demonstrating the power of abstraction. It can be connected to other methodologies, for instance vectors and matrices: set $C = (A, B)$ and the model becomes

$$C_0 = (6, 9)$$
$$C_1 = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 1 \end{pmatrix} C_0$$
$$C_2 = \begin{pmatrix} 1 & 1/3 \\ 0 & 2/3 \end{pmatrix} C_1$$

Multiplying the coefficient matrices gives a direct description of output from input and enables exploration of the relationship. Is there an initial distribution that leads to exactly the same final distribution? In another direction, one can also see how a large number of "players" could give a cellular automaton.

Finally, modeling can be a rich activity even when students cannot analyze the model. For instance as soon as 'rate of change' is introduced they could model physical systems as differential equations, and then see computer graphs of solutions. Or they might be motivated to learn relevant analytic techniques.

2.3. **Interference from customary usage of parentheses.** Grouping of subexpressions is an essential part of the structure of most mathematical expressions. Further, parsing expressions should follow this structure from the outside in: locate outermost groups and their relationships, then find immediate subgroups of these, and so on down to indivisible components.

Customary usage interferes cognitively with mathematical work in two ways:

- The customary parsing order used in reading (left to right in English) is almost always different from mathematical parsing order.
- The customary parenthesis notation does a bad job representing grouping: the opening '(' and closing ')' of a group are mathematically connected, but

they have to be found by preliminary parsing (usually using reading order) because they are not notationally connected.

Current practice in elementary education is to avoid the issue by avoiding the use of grouping, and largely sticking with reading parsing order. This has unfortunate consequences:

- Most expressions cannot be written without grouping notation, so scope is very limited.
- On-the-fly arithmetic is often necessary to avoid intermediate expressions that require grouping. The task-separated multiplication algorithm used in §2.1, for instance, requires extravagant use of parentheses. As a result the cognitive costs of task-switching cannot be avoided.
- Parenthesis-avoidence is embedded in goals: the customary meaning of "simplify" is essentially "find an equivalent expression without parentheses". This interferes with more intelligent goals in later work.
- Students do not learn how to parse non-trivial mathematical expressions.

We suggest fixing the notation rather than avoiding it. In 3.1.4 the underbrace used in §2.1.2 to indicate outcomes of evaluation also connects parentheses. This is not a good general solution because the underbrace is a powerful way to indicate subexpressions being manipulated, and these subexpressions usually do not correspond to parentheses. A better approach would be to join matching parentheses with an underline:

$$A + B\,\underline{\left(C - D\,\underline{(E+F)}\right)}\,.$$

This seems to address the problems, and feeds directly into underbrace processing illustrated above.

It is quite easy to design experiments to probe the effects described above. However no experiment is suggested here because this is a complex issue, and we have not identified a key or especially revealing special case.

2.4. **Interference from customary integer names.** The English name for 513 is "five hundred thirteen". This might be shortened to "five thirteen", but not to "five one three". It seems likely this interferes with mental arithmetic in two ways: first through overhead in translating 13 to "thirteen" and back, and second because "thirteen" is a cognitive unit that has to be disassociated into two digits for arithmetic processing.

The proposal is to see if the use of "math names" for integers to reduce cognitive overhead associated with customary names improves modest mental addition. The math name is simply the sequence of names of the digits: 513 is "five one three" for example. The other novelty is to use short-term persistence of verbal working memory to store the running total, to reduce interference with single-digit operations.

This trial requires transparently internalized single-digit addition; see the comment in the experiment description.

2.4.1. *Example.* To do the addition $367 + 12 + 57$ do the following:

(1) say "three six seven" out loud, to read it into verbal working memory;
(2) next add the 1 digit in 12 to the running total. Unless there is an overflow this changes only the 10^1 digit, so the new total will be "three, (new digit), seven". Begin by saying "three", then start with the 6 from the running

total, add 1 and say "seven", and finish with "seven" from the running total. The digits said out loud are the new running total.

(3) next add the 2 in 12. This usually changes only the 10^0 digit in the running total so first say "three seven" from the current total, start with the 7 from the running total, count up 2 and say "nine" out loud.

(4) now proceed to the 5 digit in 57. The verbal running total is "three seven nine" and the first task is to add 5 to the 7 in the 10^1 digit. There are two possibilities. If you see $7 + 5$ will produce an overflow then increment the next digit to the left and say "four". If awareness of impending overflow is not this transparent then begin with "three", think '$7 + 5 = 12$' and then overwrite the "three" by saying "**four**". In either case the next out-loud digit is "two", followed by "nine" from the running total.

(5) finally add the 7 digit in 57. Current total is "four two nine". Begin with "four" from the total, and deal with overflow as above.

In this context the interference-reducing strategies can be made more explicit. First, the digits are added one at a time, so using math names avoids conflict with common names that combine digits. Second, in most people verbal short-term memory is distinct from the working memory used for single-digit operations (see comment below in 'Long-term goals', however). It may take practice to access it independently. For instance in the second step above, after the addition one must recall the final digit 'seven' that was stored before the operation. It might help to think 'what was the final digit I heard myself say?'

Another helpful learning strategy is to refresh the running total between major steps. For instance the operation $367 + 12$ ends saying "three seven" and then "nine" while adding the 2 digit. Repeating "three seven nine" before beginning the next step helps prevent erosion during preparation for the next step.

2.4.2. *Experiment.* The plan is to compare accuracy and neural activity of mental addition using customary methodology, and with the method described above.

Subjects should be screened for automatic facility with single-digit sums, done in a way that does not require short-term verbal memory (see below

In the first imaging trial subjects are asked to do tasks mentally (no external working memory) using customary methodology. Tasks (described below) are presented visually and answers are given verbally. There are no time constraints and they are asked to be as accurate as possible.

Subjects are then taught the reduced-interference procedure described above, and practice enough to become reasonably proficient. The imaging trial is then repeated with this methodology.

2.4.3. *Outcomes, and task selection.* The reduced-interference version should enable significantly higher accuracy for some problem types. For instance with practice it should be possible to start with a four-digit integer and add ten two-digit integers, a feat that is quite hard to do accurately with traditional methods.

Tasks should be designed so the two methods have clear differences in outcomes and imaged activity. With high-school or beginning-college students it seems likely that adding three 2-digit integers to a 3-digit integer (e.g. $367+12+57+24$) would do this, but task design should be explored with preliminary trials.

2.4.4. *Long-term goals.* The real question is if using 'math names' for integers from the very beginning would make arithmetic considerably more accessible for young children. The proposal concerns an analogous task for an age group that is easier to work with, as a starting point. This analogous task (mentally adding ten two-digit integers to a four-digit integer) is not itself a high-priority skill.

I describe an example because it relates to screening for the trial. A strategy for single-digit addition is to count up using short-term verbal memory as a counting register. For instance to add 3 to the 10^1 digit of 567, start with "five six seven" and increment three times: "five seven seven", "five eight seven", "five nine seven". But doing single-digit addition this way would conflict with use of verbal memory for the running total in multi-digit addition, so subjects who do this should be disqualified.

This is the meaning of screening for "automatic facility". Even people who learn single-digit addition this way should eventually internalize it in ways that leaves verbal memory available, just as most people who learn to read by sounding out words eventually stop actually making sounds. It may involve *imagining* counting out loud, but use of a 'mental voice' seems not to overwrite actual vocalization.

3. Subliminal learning and reenforcement

Human brains are complex, and the relative lack of integration in childrens' brains means early learning has additional complexity. The fact is well-known but many of the details are invisible to adults. The proposals concern subliminal learning of algebraic structure in by-hand arithmetic, and reenforcement of qualitative geometric structure in by-hand graphing of functions. Both of these are usually lost in calculator-oriented programs. The goal is to understand these well enough to design programs in which subliminal learning and technology can coexist.

3.1. **Subliminal algebra in integer multiplication.** The first part of the experiment compares multiplications done by hand and with a calculator. This is to establish bases for comparison in the second part, and to compare the by-hand activity with algebraic manipulation. The second part compares two versions of a task-separated algorithm: one by hand, and one with primitive computational support. See the discussion for explanation.

3.1.1. *Experiment, part one.* Subjects should be high school or beginning college students, with reasonable facility with both calculators and hand arithmetic.

The tasks are to find 3×3-digit products (e.g. 946×735) either by hand using the method they were taught in school, or with a calculator, as directed. Answers should be written in either case. They should be told that accuracy is more important than speed.

3.1.2. *Discussion, part one.* The number of digits is chosen so by-hand work will fully engage the algorithmic structure, but not be overwhelmed by written intermediates.

Neural activity in the calculator case should be input/output and translation of digits to key presses. Little or no numerical or symbolic activity is expected. By-hand multiplication should show input-output, number-fact recall, and organizational activity. The interesting questions concern the organizational activity and errors; see the discussion for part two.

3.1.3. *Experiment, part two.* Subjects are taught to use a task-separated multiplication algorithm modeled on polynomial multiplication, and a final assembly (see below for notation and an example). The experiment has two versions:

- Use the algorithm to reduce 3×3-digit products to 1×1-digit products and additions. Carry these out by hand.
- Use the algorithm with 2-digit blocks (see 3.1.5) to reduce 6×6-digit products to 2×2-digit products and additions. Carry these out with a calculator.

3.1.4. *Single-digit algorithm.* The place-value notation describes integers as polynomials in powers of ten with single-digit coefficients. For example $946 = 9x^2 + 4x^1 + 6x^0$, evaluated at $x = 10$. The plan is to multiply using the polynomial algorithm of 2.1.2, then evaluate at powers of ten. Some care with notation is necessary.

We can avoid writing numbers explicitly as polynomials, by writing the exponent over the digit[2]. For instance to compute 946×735 write

$$\overset{210}{946} \times \overset{210}{735}$$

Next write a template for the organizational step:

$$\overset{4}{*}\underbrace{(\quad\;)} + \overset{3}{*}\underbrace{\left(\qquad\qquad\right)} + \overset{2}{*}\underbrace{\left(\qquad\qquad\qquad\right)} + \overset{1}{*}\underbrace{\left(\qquad\qquad\right)} + \overset{0}{*}\underbrace{(\quad\;)}$$

Note that parentheses are connected by underbraces that will eventually be used to indicate outcomes. The polynomial model only has the parentheses at this stage, but disconnected parentheses are problematic in elementary education (see 2.3). The notation here uses $\overset{2}{*}$ as a shorthand for 10^2, but it is not clear this is a good idea.

Collect coefficient products for each total coefficient 0–4:

$$\overset{4}{*}\underbrace{(9\cdot 7)} + \overset{3}{*}\underbrace{\left(9\cdot 3 + 4\cdot 7\right)} + \overset{2}{*}\underbrace{\left(9\cdot 5 + 4\cdot 3 + 6\cdot 7\right)} + \overset{1}{*}\underbrace{\left(4\cdot 5 + 6\cdot 3\right)} + \overset{0}{*}\underbrace{(6\cdot 5)}$$

Then do the multiplication and addition (in separate stages):

$$\overset{4}{*}\underbrace{(9\cdot 7)}_{63} + \overset{3}{*}\underbrace{\left(\underbrace{9\cdot 3}_{27} + \underbrace{4\cdot 7}_{28}\right)}_{55} + \overset{2}{*}\underbrace{\left(\underbrace{9\cdot 5}_{45} + \underbrace{4\cdot 3}_{12} + \underbrace{6\cdot 7}_{42}\right)}_{99} + \overset{1}{*}\underbrace{\left(\underbrace{4\cdot 5}_{20} + \underbrace{6\cdot 3}_{18}\right)}_{38} + \overset{0}{*}\underbrace{(6\cdot 5)}_{30}$$

Finally, assemble the pieces by writing them in offset rows and adding:

0					3	0
1				3	8	
2			9	9		
3		5	5			
4	6	3				
sum	6	9	5	3	1	0

[2]This is a compressed notation for computation, not general purpose. A free-standing monomial $9 \cdot 10^2$ should be written that way, not as $\overset{2}{9}$.

The left column contains the exponent, which is also the offset.

3.1.5. *Block algorithm.* Multiplication using 2-digit blocks begins by expressing integers as polynomials in 10^2 with 2-digit coefficients. For instance $638521 = 63x^2 + 85x^1 + 21x^0$, with $x = 100$. A 6×6-digit product thus becomes a 3×3-block product, and uses the same format as above.

Example: Use 2-digit blocks and a calculator to find 638521×997201.

As above we avoid writing explicit polynomials by spliting into blocks and recording the exponent over each block:

$$\overset{2}{63}\,\overset{1}{85}\,\overset{0}{21} \times \overset{2}{99}\,\overset{1}{72}\,\overset{0}{01}$$

Next collect coefficient products for each total coefficient 0–4, and do the coefficient arithmetic with a calculator:

$$100^4(\underbrace{63 \cdot 99}_{6237}) + 100^3\,\underbrace{(\underbrace{63 \cdot 72}_{4536} + \underbrace{85 \cdot 99}_{8415})}_{12951} + 100^2\,\underbrace{(\underbrace{63 \cdot 01}_{63} + \underbrace{85 \cdot 72}_{6120} + \underbrace{21 \cdot 99}_{2079})}_{8262} + \cdots$$

Note we are explicitly writing powers of 100 instead of the shorthand used in the single-digit case.

The final step is to assemble the pieces by writing them in offset rows and adding, as above.

3.1.6. *Discussion.* The traditional algorithm has been optimized for production use by experienced users, by minimizing the writing needed. Essentially any modification will be less efficient. But production arithmetic is no longer done by hand, so improved cognitive benefits may well justify some loss of efficiency. The goal of this experiment is to assess the cognitive benefits of the expanded algorithm.

In actual practice the efficiency/clarity tradeoff should mean that many fewer problems are assigned, but a success rate of 100% (after locating and correcting errors) would be expected. The presumption above is that single-digit multiplications would be done mentally, but see the next section for an alternative.

The two-digit block version would be used to lead students (subliminally) to separate the structural pattern from the blocks (i.e. not think of the algorithm as something special about digits). The result should be an effective template for multiplication of polynomials or other compound expressions in algebra.

Finally, advanced students, or group projects, can use the 4-digit block analog to multiply integers with 15 or more digits using ordinary calculators; see §3.1.1 of [16](a).

3.2. **Subliminal learning of number facts.** The goal is to have students learn single-digit products subliminally and in context rather than by explicit memorization.

The context is the task-separated algorithm described in §3.1.4. Students would be given a multiplication table on a card, see Figure 1, and given multiplication problems beginning with 1×1-digits and working up to 3×3. In multi-digit cases they would do the organizational step to reduce to single-digit products. Then they would do the batch of single-digit products, using the card for ones they do not remember. Remembering has payoffs in faster work and avoiding attention breaks. If cards and procedures are well-designed for subliminal assimilation then children would learn these fairly quickly and painlessly.

Behavioral studies can incrementally improve design of cards and procedures. The job of neuroscience is to guide improvements that educators will not reach by incremental changes. Examples illustrated in the card in Figure 1:

- Students should be instructed to read the entry out loud each time they use the card, to provide verbal reenforcement and because most people store multiplication facts in verbal memory.
- The entries on the card are complete segments to be read, not just the answer.
- Entries are designed for accurate recall. For instance $\times 7, 5; 35$ for $7 \times 5 = 35$ begins with the operation ($\times$) because beginning with 7 invites confusion with $7 + 5 = 12$.
- Two-digit answers should probably be read as digits rather than customary names, to avoid translation overhead (§2.4).
- "Equals" is omitted to shorten the entry and because it is redundant in context. Emphasis can be used as a substitute to clarify the separation between input and output, e.g. read $\times 7, 5; 35$ as "times seven, five; **three five**".

Finally, neuroscience studies have confirmed that incorrect internalizations quickly become very difficult to correct [5] [3]. It is therefore vital that they be found and fixed as soon as possible. To accomplish this, *every assignment* be checked for correctness, and students required to locate and correct errors in their work record. (Recall that this approach would use fewer assignments than is now the custom.) *Always* having to find errors also provides consistent reenforcement for accuracy and good work habits.

2	$\times 2, 2; 4$	$\times 2, 3; 6$	$\times 2, 4; 8$	$\times 2, 5; 10$	$\times 2, 6; 12$	$\times 2, 7; 14$	$\times 2, 8; 16$	$\times 2, 9; 18$
3	$\times 3, 2; 6$	$\times 3, 3; 9$	$\times 3, 4; 12$	$\times 3, 5; 15$	$\times 3, 6; 18$	$\times 3, 7; 21$	$\times 3, 8; 24$	$\times 3, 9; 27$
4	$\times 4, 2; 8$	$\times 4, 3; 12$	$\times 4, 4; 16$	$\times 4, 5; 20$	$\times 4, 6; 24$	$\times 4, 7; 28$	$\times 4, 8; 32$	$\times 4, 9; 36$
5	$\times 5, 2; 10$	$\times 5, 3; 15$	$\times 5, 4; 20$	$\times 5, 5; 25$	$\times 5, 6; 30$	$\times 5, 7; 35$	$\times 5, 8; 40$	$\times 5, 9; 45$
6	$\times 6, 2; 12$	$\times 6, 3; 18$	$\times 6, 4; 24$	$\times 6, 5; 30$	$\times 6, 6; 36$	$\times 6, 7; 42$	$\times 6, 8; 48$	$\times 6, 9; 54$
7	$\times 7, 2; 14$	$\times 7, 3; 21$	$\times 7, 4; 28$	$\times 7, 5; 35$	$\times 7, 6; 42$	$\times 7, 7; 49$	$\times 7, 8; 56$	$\times 7, 9; 63$
8	$\times 8, 2; 16$	$\times 8, 3; 24$	$\times 8, 4; 32$	$\times 8, 5; 40$	$\times 8, 6; 48$	$\times 8, 7; 56$	$\times 8, 8; 64$	$\times 8, 9; 72$
9	$\times 9, 2; 18$	$\times 9, 3; 27$	$\times 9, 4; 36$	$\times 9, 5; 45$	$\times 9, 6; 54$	$\times 9, 7; 63$	$\times 9, 8; 72$	$\times 9, 9; 81$

FIGURE 1. Multiplication Card

3.2.1. *Experiment.* Most of the neuroscience input for this topic will be inference from other studies (e.g. put the operation first). Experiments like the one suggested here might fine-tune the ideas, but serious evaluation must wait on classroom trials.

Subjects would be children (perhaps fourth grade) who are successful with standard arithmetic. The task is to perform the organizational step of the task-separated multiplication algorithm, and use the multiplication card to carry out the multiplication step. The addition step would be omitted. There should be enough pre-trial practice to learn the procedure but not enough to internalize the card material. Then subjects would be imaged working problems, and locating and correcting errors in incorrect problems.

The first objective is to track internalization of the table. These students will already know single-digit products in another format, but if they are instructed to use the cards (especially reading entries out loud) then they will probably internalize the new format rather than translate what they already know. Patterns in successful internalization might help refine the procedure.

The second and more important objective is to track error handling. It is well-established that internal uncertainty about correctness causes delays and unusual patterns of activity [7, 6, 4, 17]. For operational purposes we call an internalization "bad" if it is incorrect but is so firmly embedded that it does not provoke this error-related activity. It is urgent that incorrect internalizations be identified and fixed before they go bad. However, little is known about the process or the size of the window of opportunity.

- What is the repetition rate of errors during a session if error feedback is not received until the next session? How does internal error awareness change with repetition? Sessions should involve 30–40 problems for this.
- Compare this with correctness feedback and error correction after each problem.

The final question concerns durability. Durable knowledge requires practice well beyond achieving accurate performance (cognitive psychologists use the unfortunate term "overlearned" for this). It will be important to know how much reenforcement is necessary to achieve durability with this particular task. This might be addressed with followup studies, but getting reliable conclusions will be difficult: long periods of disuse will lead to serious interference from standard multiplication habits.

3.3. **Kinetic reenforcement of geometric structure.** Qualitative geometric structure is used to explore questions about functions, and to clarify the quantitative information needed for specific questions. For example the curves $y = ax^{2n}$ for a positive and n a positive integer, all have pretty much the same shape. We can see, for instance, that a straight line will intersect any of them in either two points, one point (when they are tangent), or no points.

We want to compare a purely visual approach with one that includes reenforcement. The comparison is done by cross-testing so the precise questions addressed are: how well does kinetic learning transfer to visual testing, and how well does visual learning transfer to kinetic testing. In fact actual *use* of qualitative structure requires hand drawing, so the crucial question concerns visual to kinetic transfer.

The role of neuroscience is to throw light on the mechanisms (or non-mechanisms) of transfer between domains. To what extent does learning in one mode recruit activity in regions that are used in testing the other mode? Does recruitment, or lack thereof, explain success or failure of transfer? Answering these questions requires imaging the learning activity, not just the testing.

3.3.1. *The experiment.* Subjects should be non-remedial first-year college students, as above. The study design depends on the number of subjects that can be tested.

If the number is twenty or fewer then students should be pre-tested to assess competence in the two learning modes, and assigned to the variant corresponding their strength. In other words, students from largely-visual technology programs should be in the visual track, and students from traditional programs should be in the kinetic track. There should be about the same number in each track.

If the number is significantly greater than twenty then students should still be pre-tested for reference purposes, but then assigned to tracks at random. This would allow assessment of cross-training. Do visually trained students adapt reasonably quickly to kinetic training, for instance?

Training sessions should last between 30 and 60 minutes, with at least three short quizzes to reenforce learning and familiarize students with the quiz format. It should be possible to repeat at least the first subsection if the corresponding quiz outcome is unsatisfactory. Students should be imaged during the training sessions. Students in both tracks should be able to do scratch work, and this should be recorded. See below for sample materials.

Next, students should be imaged taking quizzes, in a one or two day window at least three days after but within a week of the training session. The first quiz would be in the mode in which they were trained, to assess retention by comparison with the final quiz of the training session. The second quiz would be in the other mode, to assess transfer of learning.

Genuinely qualitative internalization should include some abstraction and provide flexibility. The later quizzes should be slightly different from the lesson materials to probe this.

3.3.2. *Discussion.* It seems likely that there will be significant differences in learning and transfer between the two modes. Quantifying this would require much more careful controls and larger numbers, but this experiment should suggest explanatory neural mechanisms that could substantially sharpen design of followups. For example:

- When kinetic students take visual tests, to what extent is the transfer internal, or external? External transfer would use visual comparison with a scratch sketch, while internal would presumably require communication between kinetic and visual regions, probably mediated by activity in the prefrontal cortex.
- When visual students take kinetic tests (i.e. are asked to draw something), does the learning transfer, or does the output look like a reproduction of a recalled visual image? (Sketches by students trained with graphing calculators are frequently reproductions of a calculator display.) How does neural activity reflect this?

If kinetic reenforcement is important for durable qualitative learning, then a long-term goal is to find ways to incorporate kinetic reenforcement in technology-based programs. This experiment should help make a start on this.

3.3.3. *Materials.* The experiment requires learning something unfamiliar but reasonably accessible. The proposal is to explore how the shape of a monomial ($y = x^n$) is modified by addition[3] of a lower-degree polynomial. This subliminally includes the qualitative similarity of the families $y = x^n$ for n even, and for n odd.

- The visual version is illustrated (as usual) with graphics generated by computer or calculator. Quizzes are visual multiple-choice.
- The kinetic version is illustrated by videos of hand drawing. Quizzes require drawing.

[3]Sketching the *product* of two function graphs is more useful and interesting, but probably too involved for use here.

The following illustrates visual lesson materials:

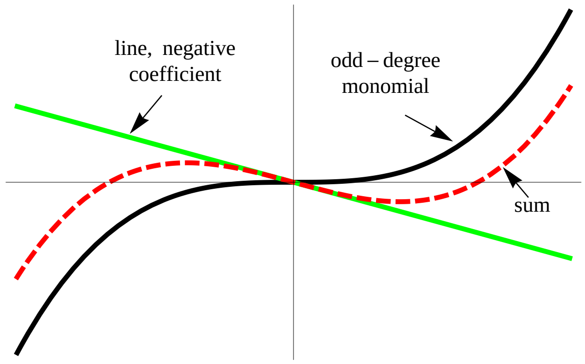

Figure 1: sum of $y = x^n$, n odd, and a line with negative coefficient.

Roughly, adding a line with negative coefficient tilts the graph a bit to the right. For very large values of x the two graphs are essentially the same.

The following illustrates a visual test item:

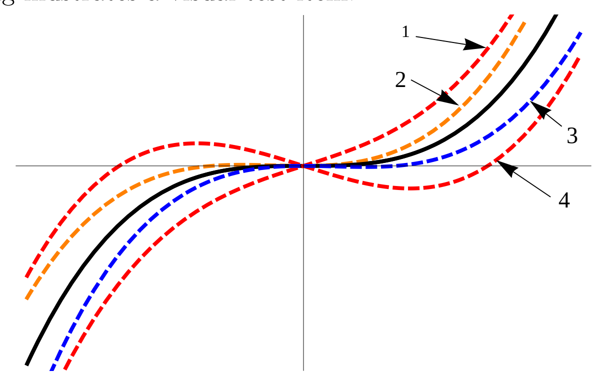

Figure 2: The solid line is the graph of a cubic monomial. Which of the functions 1–4 is the sum of this and a quadratic with negative coefficient? Which is the sum with a line with positive coefficient?

A corresponding kinetic test item would be: "sketch the graph of a cubic monomial with positive coefficient. On the same graph, sketch the sum of this and a quadratic monomial with negative coefficient."

References

[1] ACT-R: Theory and Architecture of Cognition, web site http://act-r.psy.cmu.edu/.

[2] Anderson, J.; Qin, Yulin; Sohn, Myeong-Ho; Stenger, V. A.; Carter, C. S. *An information-processing model of the BOLD response in symbol manipulation tasks* Psychonomic Bulletin and Review, **10** (2003) 241–261.

[3] Barton, Kevin; Fugelsang, Jonathan; Smilek, Daniel; *Inhibiting beliefs demands attention*, Thinking and Reasoning, **15** (2009) pp. 250–267.

[4] Carter, Cameron; van Veen, Vincent; *Anterior cingulate cortex and conflict detection: An update of theory and data*, Cognitive, Affective, and Behavioral Neuroscience **7** (2007) pp. 367–379.

[5] Dunbar, K., Fugelsang, J., and Stein, C. *Do nave theories ever go away?* In M. Lovett, and P. Shah (Eds.), Thinking with Data: 33rd Carnegie Symposium on Cognition, Erlbaum pub. (2007) pp. 193–205.

[6] Fleming, S.; Weil, R.; Nagy, Z.; Dolan, R.; Rees, G.; *Relating introspective accuracy to individual differences in brain structure*, Science **329** (2010) pp. 1541–1543.

[7] Eichele H; Juvodden H. T; Ullsperger M; Eichele T; *Mal-adaptation of event-related EEG responses preceding performance errors*, Frontiers of Human Neurosci. **4** (2010) 65.

[8] Ioannides, Andreas A. *Dynamic functional connectivity* Current Opinion in Neurobiology **17** (2007) 161170.

[9] Krueger F., Landgraf S., van der Meer E., Deshpande G., Hu X, *Effective connectivity of the multiplication network: A functional MRI and multivariate Granger causality mapping study*, Human Brain Mapping (Wiley), 2010.

[10] Krueger, F.A.; Spampinato, M. V.; Pardini, M.; Pajevic, S.; Wood, J. N.; Weiss, G. H.; Landgraf, S.; Grafman, J.; *Integral calculus problem solving: An fMRI investigation*, Neuroreport **19** (2008) 1095–1099.

[11] Liefooghe, B.; Barrouillet, P.; Vandierendonck, A.; Camos, V. *Working Memory Costs of Task Switching* Journal of Experimental Psychology: Learning, Memory, and Cognition **34** (2008) pp. 478–494.

[12] Monsell, Stephen *Task switching*, TRENDS in Cognitive Sciences Vol.7 No.3 March 2003

[13] Periáñez, J. A.; Maestú, F.; Barceó, F.; Fernández, A.; Amo, Carlos; Alonso, T. O. *Spatiotemporal brain dynamics during preparatory set shifting: MEG evidence* NeuroImage **21** (2004) pp. 687–695.

[14] Pashler, Harold, *The Psychology of Attention*, MIT Press 1998.

[15] Quinn, Frank *Towards a Science of Contemporary Mathematics*, current draft available at http://www.math.vt.edu/people/quinn/history_nature

[16] ______ *Contributions to a Science of Mathematical Learning*, preprint at http://www.math.vt.edu/people/quinn

- **a:** *Contemporary proofs for mathematics education*, to appear in the proceedings of ICMI Study 19.
- **b:** *Student computing in mathematics: interface design*
- **c:** *Mathematics Education versus Cognitive Neuroscience*
- **d:** *K-12 Calculator woes*, Notices of the Amer. Math. Soc. May 2009, p. 559.
- **e:** *Cognitive neuroscience and mathematics*

[17] Ravizza, S. M.; Anderson, J. R.; Carter, C. S. *Errors of mathematical processing: The relationship of accuracy to neural regions associated with retrieval or representation of the problem state* Brain Research, **1238** (2008) pp. 118–126.

[18] Richert, R.; Robb, M.; Fender, J.; Wartella, E. *Word Learning From Baby Videos*, Archives of Pediatric and Adolescent Medicine **164** (2010) p. 432.

[19] Roseberry S; Hirsh-Pasek K; Parish-Morris J; Golinkoff RM; *Live action: can young children learn verbs from video?* Child Dev. **80** (2009) pp. 1360-1375.

[20] Rosenberg-Lee, M., Lovett, M., Anderson, J. R. *Neural correlates of arithmetic calculation strategies* Cognitive, Affective, and Behavioral Neuroscience, **9** (2009) pp. 270–285.

[21] Smith, S.M; Miller, K. L; Salimi-Khorshidi, G.; Webster, M; Beckmann, C. F; Nichols, T. E; Ramsey, J.D; Woolrich, M. W.*Network modelling methods for FMRI* NeuroImage **54** (2011) 875891

[22] Sohn, Myeong-Ho; Ursu, Stefan; Anderson, John R.; V. Stenger, Andrew; Carter, Cameron S. *The role of prefrontal cortex and posterior parietal cortex in task switching*, Proc. Nat. Acad. Sci. **97** (2000) pp. 13448–13453.

[23] Stocco, A.; Anderson, J. R. *Endogenous Control and Task Representation: An fMRI Study in Algebraic Problem Solving* Journal of Cognitive Neuroscience **20** (2008) pp. 1300–1314.

[24] Willingham, Daniel, *Ask the Cognitive Scientist: Have Technology and Multitasking Rewired How Children Learn?*, American Educator, Summer 2010.