frontiers in Neurorobotics

BRIEF RESEARCH REPORT
published: 28 May 2021
doi: 10.3389/fnbot.2021.670895

Optimum Integration Procedure for Connectionist and Dynamic Field Equations

Andrés Rieznik[1,2]*, *Rocco Di Tella*[3,4], *Lara Schvartzman*[4] *and Andrés Babino*[5]

[1] *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Buenos Aires, Argentina,* [2] *INCYT, CONICET-INECO, Buenos Aires, Argentina,* [3] *El Gato y La Caja, Buenos Aires, Argentina,* [4] *Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires (UBA), Buenos Aires, Argentina,* [5] *Integrative Neuroscience Lab, The Rockefeller University, New York, NY, United States*

Connectionist and dynamic field models consist of a set of coupled first-order differential equations describing the evolution in time of different units. We compare three numerical methods for the integration of these equations: the Euler method, and two methods we have developed and present here: a modified version of the fourth-order Runge Kutta method, and one semi-analytical method. We apply them to solve a well-known nonlinear connectionist model of retrieval in single-digit multiplication, and show that, in many regimes, the semi-analytical and modified Runge Kutta methods outperform the Euler method, in some regimes by more than three orders of magnitude. Given the outstanding difference in execution time of the methods, and that the EM is widely used, we conclude that the researchers in the field can greatly benefit from our analysis and developed methods.

Keywords: cognitive neuroscience, connectionism and neural nets, neurorobotic, cognitive models, numerical optimisation techniques

OPEN ACCESS

Edited by:
Di Wu,
Chongqing Institute of Green and Intelligent Technology (CAS), China

Reviewed by:
Yi He,
University of Louisiana at Lafayette, United States
Yusuf Jikantoro,
Ibrahim Badamasi Babangida University, Nigeria

****Correspondence:***
Andrés Rieznik
arieznik@gmail.com

Received: *22 February 2021*
Accepted: *20 April 2021*
Published: *28 May 2021*

Citation:
Rieznik A, Di Tella R, Schvartzman L and Babino A (2021) Optimum Integration Procedure for Connectionist and Dynamic Field Equations.
Front. Neurorobot. 15:670895.
doi: 10.3389/fnbot.2021.670895

INTRODUCTION

Recent research indicates that connectionist and dynamic field theory (DFT) models might push developments in various branches of robotics (Schürmann et al., 2019; Tekülve et al., 2019; Schürmann and Beckerle, 2020; Torricelli et al., 2020) and specifically in embodied artificial cognitive systems (Lomp et al., 2016). Optimum integration procedures for these models may greatly contribute to the development of proper architectures, by accelerating simulation times, or equivalently, by reducing numerical errors.

For instance, when DFT architectures are used in artificial cognitive systems that are tied to real sensory data and drive autonomous robots, the relation between the physical time, when the computer provides a new value for the dynamical variables, and the simulated time is important. Ideally, computation time is not a concern because the computer is fast enough to provide updates within the time interval that is an adequate time step for the dynamics with the desired time scales. But, if the computer systematically takes longer to provide an update of the dynamic variables than the simulated time, then the dynamics cannot be realized on the artificial cognitive system. Open software frameworks as *cedar* typically address these cases by decreasing the simulation time at the next update, by increasing the integration time step, to bring the simulation time back in line with physical time. But given that DFT architectures can become large and complex, potentially including dozens of fields of different dimensionality, this procedure may lead to prohibitively large errors (Lomp et al., 2016).

Another example comes from the field of mathematical cognition, where a well-known non-linear connectionist model is used to simulate response times and error patterns in single-digit multiplication (Verguts and Fias, 2005). This model explains why, for instance, universally the most common mistake when retrieving 7×8 is to answer 54 instead of 56. In order to verify that this model produces observed pattern of human responses, thousands of simulations with different input values and noise levels are required. Also, if this model, as suggested in recent articles (Campbell et al., 2015; Bellon et al., 2016), should be extended or modified to explain phenomena such as retrieval-induced interference between multiplications and additions (Campbell et al., 2015) or inhibition related to individual differences (Bellon et al., 2016), it is clear that the exploration of different tentative architectures require optimum integration procedures to be practical, taking reasonable simulation times.

To the best of our knowledge, in order to simulate connectionist and dynamic field models the Euler Method (EM) is widely used. It was used, for instance, in Thelen et al. (2001) (as explicitly stated in page 21, second paragraph), in Verguts and Fias (2005) (according to a personal communication), in Lomp et al. (2016), and in Tekülve et al. (2019). In Lomp et al. (2016), the authors explicitly stated that they use the EM and did not use higher-order numerical methods because they require very many function evaluations per time step, which defeats their computational advantage when each evaluation is computationally costly.

Here we introduce two alternative numerical methods (a modified version of the fourth-order Runge Kutta method, and one semi-analytical method) and apply them to solve a well-known non-linear connectionist model of retrieval in single-digit multiplication (Verguts and Fias, 2005). We show that, in many regimes, the semi-analytical and modified Runge Kutta methods outperform the Euler method, in some regimes by more than three orders of magnitude. Given the difference in execution time of the methods, and that the EM is widely used, we conclude that the researchers in the field can greatly benefit from our analysis.

METHODS

In the connectionist and dynamic field frameworks, the propagation of activation is a dynamic process operating in a continuous state space and evolving continuously over time (Grossberg, 1982; Thelen et al., 2001; Munakata and McClelland, 2003). In models that simulate this gradual activation process, it is typically formalized as a differential equation relating the rate of change of some variable (such as the activation of some unit) to the inputs it is currently receiving from other units via weighted connections. Equations describing the dynamic behavior of each unit are of the form

$$\dot{u}\,(t) = a\,(t) * u\,(t) + b\,(t) \equiv F\,(u, t) \qquad (1)$$

where $u(t)$ describes the activation of one unit. a(t) and $b(t)$ are weighted sums over all other units, governed each one by an equation of this form. $a(t)$ also contains a spontaneous decay term, i.e., $a(t) = -1 + \ldots$ In **Appendix A** we discuss how connectionist and dynamic field equations can be reorganized to have the form of Equation (1) using two specific examples. We take advantage of the fact that both models have this form to develop optimum integration procedures that can potentially be applied to integrate both models, since the procedures we present here can in principle be applied to any set of coupled equations with the form given by Equation (1).

On the one hand, the Euler Method (EM) to solve Equation (1) is given by

$$u\left(t + dt\right) = u\,(t) + F\,(u, t) * dt \qquad (2)$$

where dt is the integration time-step.

On the other hand, to apply the fourth-order Runge-Kutta method (RK4M) to solve Equation (1) we need to calculate the coefficients $a(t)$ and $b(t)$ many times at each step. Since these coefficients are weighted sums over all other units, the bulk of the computing time is expended on these terms, at a great computational cost. The improved accuracy by the application of the RK4M comes with the cost of computing $a(t)$ and $b(t)$ to obtain k2, k3, and k4. To avoid this calculation, in the modified version we apply the original algorithm to calculate k1, but we only update the value for $u(t)$ for the calculation of k2, k3, and k4. If, in Equation (1), we explicitly write the dependency of F on $a(t)$ and $b(t)$, the modified RK4M obeys

$$\begin{aligned}
k1\,(t) &= F[t, u(t), a(t), b(t)] \\
k2\,(t) &= F[t + \frac{dt}{2}, u\,(t) + k1{*}\frac{dt}{2}\,, a(t), b(t)] \\
k3\,(t) &= F\left(t + \frac{dt}{2}, u\,(t) + k2{*}\frac{dt}{2}, a\,(t)\,, b\,(t)\right) \\
k4\,(t) &= F[t + dt, u\,(t) + k3{*}dt, a(t), b(t)]
\end{aligned}$$

and, as usual, $u(t + dt) = u(t) + dt^{*}(\mathrm{k1} + 2\mathrm{k2} + 2\mathrm{k3} + \mathrm{k4})/6$.

Finally, for the derivation of the semi-analytical method (SAM), note that, in the particular case in which $a(t) = cte = a$ and $b(t) = cte = b$, Equation (1) has analytical solution:

$$f\left(t + dt\right) = \left(f\,(t) + \frac{b(t)}{a(t)}\right) * e^{a(t)\,*dt} - \frac{b(t)}{a(t)} \qquad (3)$$

Even though $a(t)$ and $b(t)$ are not constants, the implementation of this "analytical" solution at each step of integration converges to the solution when dt goes to zero. One of us (AR) used a similar approach to study optical amplifiers' dynamic behavior (Rieznik and Fragnito, 2004), also governed by equations of the form of Equation (1). The analogy with an amplified signal can be more easily understood by rearranging Equation (3) as

$$f\left(t + dt\right) = f\,(t)\; *G\,(t) + \frac{b\,(t)}{a\,(t)}(G\,(t) - 1) \qquad (4)$$

where $G(t) = \exp[a(t)^{*}dt]$ is the signal gain. In Equation (4), the first term of the right-hand side represents the input activation multiplied by the gain or absorption that all other units produce on the activated unit. The second term is the noise term, named this way because it is present even in the absence of an input signal, and it is not modulated by this input. In fact, using this analogy, the effects of all other units on the activation of one given unit can be divided into two contributions: the noise contribution and the contribution that amplifies or absorbs the input signal.

We implement the SAM, the EM, and the modified RK4M to solve the IN model equations (Verguts and Fias, 2005), described in **Appendix B**. IN is a connectionist model of retrieval in single-digits multiplications, an area of interest for our research group (Zimmerman et al., 2016; Rieznik et al., 2017). The most important facts of arithmetic retrieval are (Verguts and Fias, 2005; Zimmerman et al., 2016): the problem size effect (small multiplication problems are easier than larger ones; cf. 3×2 and 7×8), the five effect (problems with five are easier than can be accounted for by their size), and the tie effect (problems with identical operands are easier than other problems; cf. 8×8 and 8×7). The basic assumption is that candidate answers to a particular problem are in cooperative/competitive interactions, and these interactions favor small, five, and tie problems.

RESULTS

To compare the three methods, we simulate the activation patterns of the network for the input 7×8, the most difficult operation in the multiplications table. The IN model describes a total of 107 coupled equations (see **Appendix B**): eight for each of the two-input fields (varying from 2 to 9), 36 semantic fields, nine tens fields (from 0 to 8), 10 units fields (from 0 to 9), and 36 response fields. To measure the accuracy of a simulation, we compute an average error over the response fields: if u_n is the numerical solution and u_r is the real one, the error is given by

$$error = sum(|u_n - u_r|)/36 \tag{5}$$

Observe that the sum is performed over the 36 response fields, which explains why we divide by 36 to compute the error. Here, the real solution u_r is obtained using a very small dt (in our case, 10^{-9}). We checked that the three methods converge to the same u_r solution.

We use the following input parameters: $B = 20$, $M = 10$, $S = 2$, tao $= 19.5$, alpha $= 0.75$, and $C = 0.5$. It was shown in Verguts and Fias (2005) that these parameters fit human performance of retrieval in single-digits multiplications. All 107 input fields were zero at $t = 0$, except the first-operand input field for 7 and the second-operand input field for 8, which values were 20 ($B = 20$). After $t = 0$ it is assumed that the input stimuli are no longer present. The simulation is stopped at $t = 0.1$ (a.u.), which is the time that, for these parameters, it takes to the model to provide an answer (56, i.e., the result of 7×8). All simulation durations, given in seconds, were taken in a Yoga-Lenovo personal computer with an i7 intel processor running Matlab 2008 for windows. The Matlab codes are available at https://github.com/arieznik/IN_simulations (see **Appendix C** for instructions). Results are shown in **Figure 1**. In order to vary the simulation times and errors, we vary dt from 10^{-4} to 10^{-7}. The exponent j defining the step size ($dt = 10^{-j}$), was varied in steps of size 1 from 4 to 7 and for each of these values we also run two extra simulations, one with $dt = 0.5{*}10^{-j}$ and the other with $dt = 0.25{*}10^{-j}$.

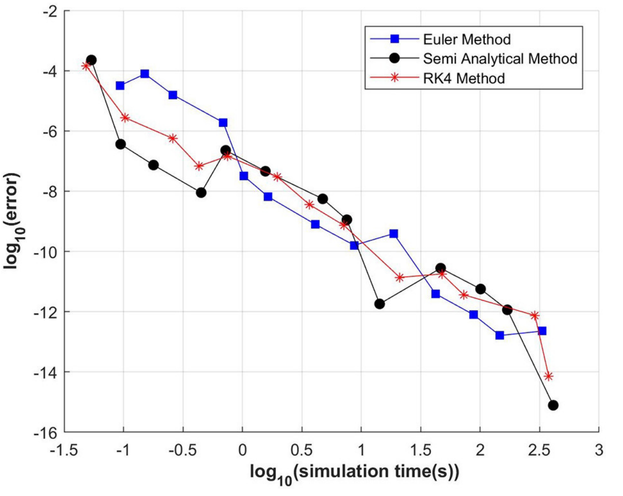

FIGURE 1 | Simulation times vs. error for three methods: EM, SAM, and modified RK4M. Results varying dt from 10^{-4} to 10^{-7} are shown.

It can be observed, in **Figure 1**, differences larger than three orders of magnitudes among the methods, but no method is the most efficient in all regimes. For any required error, the SAM or EM are faster than the RK4, in most cases by more than one or even two orders of magnitude. Equivalently, they are more precise given any required simulation time. Importantly, for commonly used errors, larger than $\sim 10^{-8}$, the EM method is overperformed by the SAM methods in some cases by more than three orders of magnitude.

DISCUSSION

The fact that connectionist and DFT models can be written in the form of Equation (1) suggests that these results could be generalized to the simulations of other similar models, simpler or more complex. In the case here analyzed there are up to three orders of magnitude of difference in execution time among the models, but, despite that, the EM is the one that it is, to the best of our knowledge, universally used. Researchers in the field can benefit from these results: the two methods we introduce here can be easily implemented in open software like *cedar*; the matlab codes we share in our github repository (https://github.com/arieznik/IN_simulations) are well-commented and easy to follow in order to facilitate other researchers' further investigations on the regimes under which these methods overperform the EM in areas which are not the areas we are interested in (the interacting neighbors model for mathematical cognition).

These outstanding differences in simulation times using different methods also pose a question: how could a naïve user of the methods decide which one to use? Step-doubling is a well-known technique to estimate the local error and could be used to choose among methods (Gear, 1971).

In modeling optical amplifiers, adaptive step size selection criteria are used (Sinkin et al., 2003; Heidt, 2009; Rieznik et al., 2012). The integration step size (dt in this paper) varies along with the simulation according to a required-by-the-user local error. Step-doubling and linear extrapolation is used to obtain the higher-order solution (Gear, 1971). To the best of our knowledge, although widely used in other fields, these techniques have not been previously used, or even seriously investigated in simulations of the connectionist and DFT frameworks with applications in robotics. Although adaptive step size is not suitable when the evolution in time of sensory readings must be monitored, it may greatly improve simulations performances in other areas of robotics research. Under this framework, the user sets a target error and not a fixed step size.

In conclusion, we presented preliminary results suggesting that the two alternative methods to the EM for the integration of connectionist and dynamic field models (the SAM and modified RK4M that we developed here) may greatly improve the simulations, but further investigation is necessary to understand the regimes under which the application of the SAM or RK4M is preferable over the EM, which is, to the best of our knowledge, the default method used by researchers.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

AR: project supervision and methodology. AR, RD, LS, and AB: materials development, software development, and validation. AR and AB: data reduction, data analysis, and writing. RD: writing. All authors contributed to the article and approved the submitted version.

REFERENCES

Bellon, E., Fias, W., and De Smedt, B. (2016). Are individual differences in arithmetic fact retrieval in children related to inhibition? *Front. Psychol.* 7:825. doi: 10.3389/fpsyg.2016.00825

Campbell, J. I., Dufour, K. D., and Chen, Y. (2015). Retrieval-induced forgetting of multiplication facts and identity rule. *Mem. Cogn.* 43, 672–680. doi: 10.3758/s13421-014-0483-1

Gear, C. W. (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*. Hoboken, NJ: Prentice Hall PTR.

Grossberg, S. (1982). *How Does a Brain Build a Cognitive Code? Studies of Mind and Brain*. https://www.google.com/search?sxsrf=ALeKk03MbXIJNe_7CMioeODvv-hjxqwsCA:1619393101847&q=Berlin&stick=H4sIAAAAAAAAAAOPgE-LUz9U3MImvKjFQ4gAxDU3NCrVUs5Ot9POL0hPzMqsSSzLz81A4Vmn5pXkpqSmLWNmcUotyMvN2sDICADsBdudKAAAA&sa=X&ved=2ahUKEwjvgtm0xZrwAhU94XMBHbY0BUQQmxMoATAnegQILBAD Berlin: Springer, 1–52.

Heidt, A. M. (2009). Efficient adaptive step size method for the simulation of supercontinuum generation in optical fibers. *J. Light. Technol.* 27, 3984–3991. doi: 10.1109/JLT.2009.2021538

Lomp, O., Richter, M., Zibner, S. K., and Schöner, G. (2016). Developing dynamic field theory architectures for embodied cognitive systems with cedar. *Front. Neurorobot.* 10:14. doi: 10.3389/fnbot.2016.00014

Munakata, Y., and McClelland, J. L. (2003). Connectionist models of development. *Dev. Sci.* 6, 413–429. doi: 10.1111/1467-7687.00296

Rieznik, A., Lebedev, M., and Sigman, M. (2017). Dazzled by the mystery of mentalism: the cognitive neuroscience of mental athletes. *Front. Hum. Neurosci.* 11:287. doi: 10.3389/fnhum.2017.00287

Rieznik, A. A., and Fragnito, H. (2004). Analytical solution for the dynamic behavior of erbium-doped fiber amplifiers with constant population inversion along the fiber. *J. Opt. Soc. Am. B* 21, 1732–1739. doi: 10.1364/JOSAB.21.001732

Rieznik, A. A., Heidt, A. M., Konig, P. G., Bettachini, V. A., and Grosz, D. F. (2012). Optimum integration procedures for supercontinuum simulation. *IEEE Photon. J.* 4, 552–560. doi: 10.1109/JPHOT.2012.2188281

Schürmann, T., and Beckerle, P. (2020). Personalizing human-agent interaction through cognitive models. *Front. Psychol.* 11:561510. doi: 10.3389/fpsyg.2020.561510

Schürmann, T., Mohler, B. J., Peters, J., and Beckerle, P. (2019). How cognitive models of human body experience might push robotics. *Front. Neurorobot.* 13:14. doi: 10.3389/fnbot.2019.00014

Sinkin, O. V., Holzlohner, R., Zweck, J., and Menyuk, C. R. (2003). Optimization of the split-step Fourier method in modeling optical-fiber communications systems. *J. Light. Technol.* 21, 61–68. doi: 10.1109/JLT.2003.808628

Tekülve, J., Fois, A., Sandamirskaya, Y., and Schöner, G. (2019). Autonomous sequence generation for a neural dynamic robot: scene perception, serial order, and object-oriented movement. *Front. Neurorobot.* 13:95. doi: 10.3389/fnbot.2019.00095

Thelen, E., Schöner, G., Scheier, C., and Smith, L. B. (2001). The dynamics of embodiment: a field theory of infant perseverative reaching. *Behav. Brain Sci.* 24, 1–34. doi: 10.1017/S0140525X01003910

Torricelli, D., Rodriguez-Guerrero, C., Veneman, J. F., Crea, S., Briem, K., Lenggenhager, B., et al. (2020). Benchmarking wearable robots: challenges and recommendations from functional, user experience, and methodological perspectives. *Front. Robot. AI* 7:168. doi: 10.3389/frobt.2020.561774

Verguts, T., and Fias, W. (2005). Interacting neighbors: a connectionist model of retrieval in single-digit multiplication. *Mem. Cogn.* 33, 1–16. doi: 10.3758/BF03195293

Zimmerman, F., Shalom, D., Gonzalez, P. A., Garrido, J. M., Heduan, F. A., Dehaene, S., et al. (2016). Arithmetic on your phone: a large scale investigation of simple additions and multiplications. *PLoS ONE* 11:e0168431. doi: 10.1371/journal.pone.0168431

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX

Appendix A: Reorganizing connectionist and dynamic field theory equations as Equation (1)

We show next two examples on how to reorganize model equations in order to obtain the form of Equation (1). The two examples are one connectionist model (Verguts and Fias, 2005) and one dynamic field model (Thelen et al., 2001). We begin by the later.

Equation (8) in Thelen et al. (2001) is

$$\tau \dot{u}(x,t) = -u(x,t) + S(x,t) + \int w(x-x') f(u(x'))\, dx' + h + q\xi(x,t) \tag{A1}$$

If we define $a(x,t) = -1/\tau$ and

$$\tau b(x,t) = S(x,t) + \int w(x-x') f(u(x'))\, dx' + h + q\xi(x,t) \tag{A2}$$

and we drop the x dependence, (A1) is

$$\dot{u}(t) = a * u(t) + b(t) \tag{A3}$$

which has the form of Equation (1), as we want to demonstrate.

Note that the Equation (A3) is linear because not $a*\mathrm{u}(t)$ nor $b(t)$ contain non-linear terms [$f(u)$ in Equation (A2) is a step-function which equals 1 or 0 depending on the u value]. The example we give next, however, the connectionist IN model equations, is non-linear, since, as we shall see, the term $a(t)*u(t)$ contain non-linearities. The IN model equations are explained in **Appendix B**. For the semantic fields, Equation (B2), we have

$$\frac{\boldsymbol{d}}{\boldsymbol{dt}} y_i(t) = -y_i(t) + \sum_j w_{ji}^{in} x_j \left(B - y_i(t)\right) \tag{A4}$$

which can be reorganized as

$$\frac{\boldsymbol{d}}{\boldsymbol{dt}} y_i(t) = -y_i(t)\left(1 + \sum_j w_{ji}^{in} x_j\right) + \sum\nolimits_j w_{ji}^{in} x_j B \tag{A5}$$

Now, defining

$$a_i(t) = -\left(1 + \sum_j w_{ji}^{in} x_j\right)$$

and

$$b_i(t) = \sum\nolimits_j w_{ji}^{in} x_j B,$$

And dropping the subscript i, Equation (A4) can be rewritten as

$$\dot{y}(t) = a(t) * y(t) + b(t)$$

TABLE A1 | Coefficients $a(t)$ and $b(t)$ for each field in the IN model equations.

(Equation) Field	a_i(t)	b_i(t)
(B1) input field	$-(1 + I_i(t))$	$BI_i(t)$
(B2) semantic field	$-\left(1 + \sum_j w_{ji}^{in} x_j(t)\right)$	$B\sum_j w_{ji}^{in} x_j(t)$
(B3) decade field	$-\left(1 + M\sum_j w_{ji}^{T} y_j(t) + \sum_{j\neq i} z_j^T(t)\right)$	$BM\sum_j w_{ji}^{T} y_j(t)$
(B4) unit field	$-\left(1 + M\sum_j w_{ji}^{U} y_j(t) + \sum_{j\neq i} z_j^U(t)\right)$	$BM\sum_j w_{ji}^{U} y_j(t)$
(B5) response field	$-\left(1 + \sum_j w_{ji}^{R} z_j(t)\right)$	$B\sum_j w_{ji}^{R} z_j(t)$

Which, as we want to demonstrate, has the form of Equation (1). Observe that $a(t)*\mathrm{y}(t)$ contains non-linearities, since $a(t)$ is linear on the other units and then $a(t)*\mathrm{y}(t)$ contains the multiplication of two units' values. In **Table A1** we show the expressions for $a(t)$ and $b(t)$ for the other fields in the IN model.

Appendix B: IN model equations

Note: the equation here shown had some typos in the original article in which they were presented (Verguts and Fias, 2005). The authors sent us, through a personal communication, a corrected version, which we explain next.

For each of the two input fields, for an input unit i at time t, activation xi of that unit at time t is described by

$$\frac{d}{dt} x_i(t) = -x_i(t) + I_i(t)(B - x_i(t)) \tag{B1}$$

In Equation (A1), $-x_i(t)$ functions as a decay term. The factor $I_i(t)$ is an indicator function. which, at time t, equals one if number i was presented in its input field and zero otherwise. The parameter B is an upper bound to the activation values x_i. Hence, the combination of the decay term and the $I_i(t)[\mathrm{B} - x_i(t)]$ terms restrict x_i between zero and B.

The activation y_i of a semantic unit i has the following equation:

$$\frac{d}{dt} y_i(t) = -y_i(t) + \sum\nolimits_j w_{ji}^{in} x_j \left(B - y_i(t)\right) \tag{B2}$$

In Equation (B2), the sum is taken over all nodes in both input fields. Again, the factor B – y_i bounds the equation below B. The equation is also bounded above zero: For example, in the absence of any input, the equation becomes $\mathrm{d}y_i(t)/\mathrm{dt} = -y_i(t)$, and y_i decays exponentially toward zero. The connection weight $w^{in}{}_{ji}$ are given by $w^{in}{}_{ji} = \exp[-\alpha * \mathrm{abs}(i–j)] - \mathrm{C}$, where α and C are two input parameters for the model.

The equation describing activation $z_i{}^{\mathrm{T}}$ of unit i in the decade field (with superscript T for tens, or decades) is

$$\frac{\mathbf{d}}{\mathbf{dt}} z_i^T(t) = -z_i^T + M\sum\nolimits_j w_{ji}^T y_j \left(B - z_i^T(t)\right) - z_i^T \sum\nolimits_{j\neq i} z_j^T \tag{B3}$$

and similarly, for the units field, with superscript U for units,

$$\frac{\mathbf{d}}{\mathbf{dt}}z_i^U\left(t\right)=-z_i^U+M\sum\nolimits_j w_{ji}^U y_j\left(B-z_i^U\left(t\right)\right)-z_i^U\sum\nolimits_{j\neq i}z_j^U \tag{B4}$$

The weights ${w_{ji}}^T$ and ${w_{ji}}^U$ are zero or one depending on whether node j should feed into decomposition field unit i or not. For example, the weight from the 24-node in the semantic field to the twenties node in the decades field is one, to the thirties node it is zero, and so on.

In the response field, for node i with activation $r_i(t)$,

$$\frac{\mathbf{d}}{\mathbf{dt}}r_i\left(t\right)=\ -r_i\left(t\right)+\sum\nolimits_j w_{ji}^R z_j\left(B-r_i\left(t\right)\right) \tag{A5}$$

The weights ${w_{ji}}^R$ are zero or S, again depending on whether the connection between the decomposition (decade or unit) node and the response node should be formed or not. The parameter S is a scaling parameter which influences the overall speed of responding in the network.

In addition to these parameters, there is also a threshold that has to be reached before a response is executed, which is denoted alpha. To sum up, the parameters used in the model are B, M, S, and alpha, in addition to the parameters α and C describing the connection weight between two semantic fields. The first four of these are scaling parameters, in the sense that changing their values changes the simulated RTs (e.g., RTs are generally slower with a higher threshold alpha), but does not change the qualitative pattern of predictions.

Appendix C: Download, installation and running the codes

You must save the compressed folder called INmodel.zip in your computer. When you extract it, five matlab files (.m) will be created inside the folder IN_model:

- IN_SAM.m, IN_EM.m, and IN_RK4.m: these functions use the SAM, EM, and RK4 methods to solve the IN equations using as inputs the first operand (op1), the second operand (op2 ≤ op1), alpha, C, and dt. Be sure of adding these functions to the matlab path before the simulation starts.
- IN_Simulation.m: this is the file you must run in order to perform the simulations. By default, it runs the simulation using op1 = 8 and op2 = 7, since in this article we simulated the pattern of activations for the input 8×7. It also uses by default the IN_SAM.m function. It is easy to change the problem to be simulated (for instance, to 8×8) or the method used for the integration since the code is well-commented.
- Figure_1.m: if you run this file, after a few seconds **Figure 1** in this article will pop put.

Apart from these .m files, a set of .mat files will be created inside the IN_model folder. These are matlab variables containing the results of the simulations using the three methods here presented for the input 8×7 and with dt varying from 10-4 to 10-7. These files are used when ran Figure_1.m in order to generate **Figure 1**. Be sure they are included in the matlab path when running this file.