On tree-based methods for (partial) differential equations

NICOLAS PRIVAULT
(joint work with Guillaume Penent and Jiang Yu Nguwi)

Stochastic branching mechanisms have been used to represent the solutions of partial differential equations in [15], [7], [10], [8], and recently extended in [6] to the treatment of polynomial nonlinearities in first order gradient terms. This talk reviews an extension of such tree-based methods to functional nonlinearities with gradients of arbitrary orders.

Consider the ODE

$$u'(t) = f(u(t)), \qquad u(0) = u_0 \in \mathbb{R}^d, \quad t \in \mathbb{R}_+, \tag{1}$$

whose solution can be expanded as

$$u(t) = u_0 + tf(u_0) + \frac{t^2}{2} f' f(u_0) + \frac{t^3}{6} f' f' f(u_0) + \frac{t^3}{6} f''[f, f](u_0) + \cdots$$

which rewrites as the sum

$$u(t) = u_0 + \sum_{\mathcal{T}} \frac{t^{r(\mathcal{T})}}{\sigma(r(\mathcal{T}))\gamma(r(\mathcal{T}))} F(\mathcal{T})$$

over the family of Butcher trees $\mathcal{T}$, see [1], [2], Chapters 4-6 of [4], and [9], based on early work of [3]. In order to solve (1), we may also write

$$u(s) = u_0 + \int_0^s u'(r)dr = u_0 + \int_0^s f(u(r))dr,$$

and more generally we can expand the derivative $f^{(l)}(u(r))$ as

$$f^{(l)}(u(r)) = f^{(l)}(u_0) + \int_0^r f(u(r)) f^{(l+1)}(u(v))dv, \qquad l \geq 1.$$

We note that the above family of equations can be rewritten as

$$c(u)(t) = c(u)(0) + \sum_{Z \in \mathcal{M}(c)} \int_0^t \prod_{z \in Z} z(u)(s)ds \tag{2}$$

where c runs through a set $\mathcal{C} := \{\text{Id}, \ f^{(l)}, \ l \geq 0\}$, of functions called *codes* and $\mathcal{M}(c)$ is defined by letting $\mathcal{M}(\text{Id}) := \{f\}$ and $\mathcal{M}(g) := \{(f, g')\}$ for g a smooth function on $\mathbb{R}_+ \times \mathbb{R}$, see [12].

Next, consider a nonlinear PDE of the form

$$\begin{cases} \partial_t u(t, x) + \dfrac{1}{2} \Delta u(t, x) + f(u(t, x)) = 0 \\ u(T, x) = \phi(x), \qquad (t, x) \in [0, T] \times \mathbb{R}. \end{cases} \tag{3}$$

Letting $v(t,x) := g(u(t,x))$, we now have

$$\partial_t v(t,x) + \frac{1}{2}\Delta v(t,x) = g'(u(t,x))\left(\partial_t u(t,x) + \frac{1}{2}\Delta u(t,x)\right) + \frac{1}{2}(\partial_x u(t,x))^2 g''(u(t,x))$$

$$= -f(u(t,x))g'(u(t,x)) + \frac{1}{2}(\partial_x u(t,x))^2 g''(u(t,x)),$$

which shows that the functions u, $\partial_x u$, $af^{(k)} \circ u$ satisfy the integral equations

$$\begin{cases} u(t,x) = \displaystyle\int_{-\infty}^{\infty} \varphi(T-t,y-x)\phi(y)dy + \int_t^T \int_{-\infty}^{\infty} \varphi(s-t,y-x)f(u(s,y))dyds \\ af^{(k)}(u(t,x)) = \displaystyle\int_{-\infty}^{\infty} \varphi(T-t,y-x)af^{(k)}(\phi(y))dy \\ \displaystyle+\int_t^T \int_{-\infty}^{\infty} \varphi(s-t,y-x) \\ \qquad \times \left(af(u(s,y))f^{(k+1)}(u(s,y)) - \frac{a}{2}(\partial_x u(s,y))^2 f^{(k+2)}(u(s,y))\right) dyds \\ \partial_x u(t,x) = \displaystyle\int_{-\infty}^{\infty} \varphi(T-t,y-x)\partial_x\phi(y)dy \\ \quad \displaystyle+\int_t^T \int_{-\infty}^{\infty} \varphi(s-t,y-x)f'(u(s,y))\partial_x u(x,y)dyds, \end{cases}$$

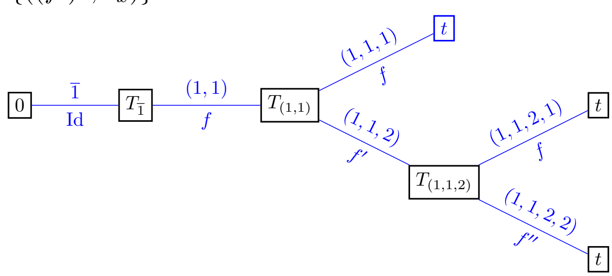
$a \neq 0$, $k \in \mathbb{N}$. We note that the above set of equations admits a formulation identical to (2) provided that we use the codes

$$\mathcal{C} := \left\{\mathrm{Id},\ \partial_x,\ af^{(k)},\ a \neq 0,\ k \in \mathbb{N}\right\}$$

and the mechanism defined as

$$\mathcal{M}(\mathrm{Id}) := \{f^*\}, \quad \mathcal{M}(g^*) := \left\{(f^*, (g')^*), \left(\partial_x, \partial_x, -\frac{1}{2}(g'')^*\right)\right\},$$

and $\mathcal{M}(\partial_x) := \{((f')^*, \partial_x)\}$.



We consider a random coding tree $\mathcal{T}_{t,x,c}$ illustrated by the above sample, started at (t,x) with a code $c \in \mathcal{C}$ and partitioned as $\mathcal{K}^\partial \cup \mathcal{K}^\circ$, where $\mathcal{K}^\circ$ denotes the set of leaves. In the next result, we use the random functional

$$\mathcal{H}(\mathcal{T}_{t,x,c}) := \prod_{\bar{k} \in \mathcal{K}^\circ} \frac{1}{q_{c_{\bar{k}}} \rho(\tau_{\bar{k}})} \prod_{\bar{k} \in \mathcal{K}^\partial} \frac{c_{\bar{k}}(u)\big(T, X^{\bar{k}}_{T_{\bar{k}}}\big)}{\overline{F}(T - T_{\bar{k}-})}.$$

of the random coding tree $\mathcal{T}_{t,x,c}$, in which branching at a node $\overline{k}$ occurs at the random time $T_{\overline{k}}$, the interjump time $\tau_{\overline{k}} = T_{\overline{k}} - T_{\overline{k}-}$ has tail CDF $\overline{F}$ and PDF ρ, and $\big(X_t^{\overline{k}}\big)_{t \geq T_{\overline{k}-}}$ is an independent Brownian motion started at time $T_{\overline{k}-}$.

Theorem 1. ([13]) *Assume that the integral solution of the system* (2) *is unique and that there exists a constant* $K > 0$ *such that:*

$$|f^{(k)} \circ \phi|_\infty \leq K, \quad k \geq 0, \quad |\phi|_\infty \leq K, \quad |\phi'|_\infty \leq K.$$

Then, there exists $T > 0$ *such that the solution of* (3) *admits the probabilistic representation*

$$u(t,x) = \mathbb{E}\left[\mathcal{H}(\mathcal{T}_{t,x,\mathrm{Id}})\right], \qquad (t,x) \in [0,T] \times \mathbb{R}.$$

The above method also extends to fully nonlinear PDEs of the form

$$\begin{cases} \partial_t u(t,x) + \dfrac{1}{2}\Delta u(t,x) + f\big(u(t,x), \nabla u(t,x), \ldots, \nabla^n u(t,x)\big) = 0, \\ u(T,x) = \phi(x), \qquad (t,x) = (t, x_1, \ldots, x_d) \in [0,T] \times \mathbb{R}^d, \end{cases}$$

$d \geq 1$, see [13], [11]. As an example, we consider a cosine nonlinearity with a gradient of order four, for which our method appears more accurate than the deep Galerkin method [14]. Related comparisons can be found in [11] with respect to the deep BSDE method [5].

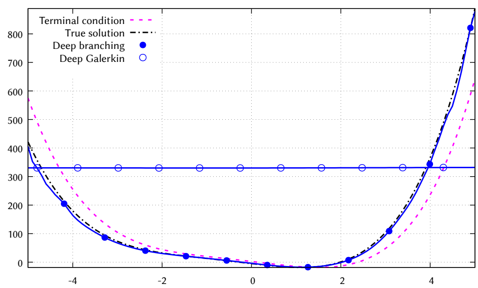


Figure 1. Comparison graphs in dimension $d = 5$.

References

[1] J.C. Butcher. Coefficients for the study of Runge-Kutta integration processes. *J. Austral. Math. Soc.*, 3:185–201, 1963.

[2] J.C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, Ltd., Chichester, third edition, 2016.

[3] A. Cayley. On the theory of the analytical forms called trees. *Philosophical Magazine*, 13(85):172–176, 1857.

[4] P. Deuflhard and F. Bornemann. *Scientific Computing with Ordinary Differential Equations*, volume 42 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2002.

[5] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[6] P. Henry-Labordère, N. Oudjane, X. Tan, N. Touzi, and X. Warin. Branching diffusion representation of semilinear PDEs and Monte Carlo approximation. *Ann. Inst. H. Poincaré Probab. Statist.*, 55(1):184–210, 2019.

[7] N. Ikeda, M. Nagasawa, and S. Watanabe. Branching Markov processes I, II, III. *J. Math. Kyoto Univ.*, 8-9:233–278, 365–410, 95–160, 1968-1969.

[8] H.P. McKean. Application of Brownian motion to the equation of Kolmogorov-Petrovskii-Piskunov. *Comm. Pure Appl. Math.*, 28(3):323–331, 1975.

[9] R.I. McLachlan, K. Modin, H. Munthe-Kaas, and O. Verdier. Butcher series: a story of rooted trees and numerical methods for evolution equations. *Asia Pac. Math. Newsl.*, 7(1):1–11, 2017.

[10] M. Nagasawa and T. Sirao. Probabilistic treatment of the blowing up of solutions for a nonlinear integral equation. *Trans. Amer. Math. Soc.*, 139:301–310, 1969.

[11] J.Y. Nguwi, G. Penent, and N. Privault. A deep branching solver for fully nonlinear partial differential equations. Preprint arXiv:2203.03234, 17 pages, 2022.

[12] G. Penent and N. Privault. Numerical evaluation of ODE solutions by Monte Carlo enumeration of Butcher series. Preprint arXiv:2201.05998, 22 pages, 2021.

[13] G. Penent and N. Privault. A fully nonlinear Feynman-Kac formula with derivatives of arbitrary orders. Preprint arXiv:2201.03882, 25 pages, 2022.

[14] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[15] A.V. Skorokhod. Branching diffusion processes. *Teor. Verojatnost. i. Primenen.*, 9:492–497, 1964.