Choosing the thermal conduction equation solution method in SPH

Vasiliy S Rykovanov and Filipp A Sapozhnikov

Russian Federal Nuclear Center — Zababikhin All-Russian Research Institute of Technical Physics (RFNC-VNIITF), Snezhinsk, Russia

E-mail: v.s.rykovanov@vniitf.ru

Abstract. The set of explicit iterative schemes for parabolic problems (heat conduction as example) is reviewed. The iterative process is constructed based on algebraic polynomials (Chebyshev, Lanczos, or Legendre) properties. They combine the simplicity of explicit time-integration with the increased allowed timestep. This approach may be applied at smoothed particle hydrodynamics (SPH) simulations as a replacement for the popular implicit time-integration schemes, whose drawbacks might get stronger especially on SPH-like contrary to mesh-style space description.

1. Introduction

In SPH method the space differential approximation is rather wide (≈30–60 neighbours in 1D, ≈50–90 in 2D, and ≈80–200 in 3D). That’s why the well-known implicit time-integration approach to the thermal conduction equation solution might become more complicated, comparing to the mesh methods with only cell boundary neighbours stencil.

On the other side, the simple and robust explicit time-integration method for parabolic equations suffer from the stability criteria for timestep \( \tau \): \( C = \frac{\tau K}{h^2} \leq \frac{1}{2} \), where \( h \) is the typical space discretization resolution, \( K \) is the so-called kinematic diffusion coefficient with dimension \([L^{-2}t]\) (\( L \) stands for unit of distance, \( t \) — unit of time).

That’s why the idea to overcome the timestep limit while retaining the explicit scheme may bring extra efficiency on massively parallel architectures, where overheads of solvers at implicit algorithms are considerable.

2. Heat conduction notation in SPH

Parabolic equation for heat conduction is

\[
\frac{\partial e}{\partial t} = \nabla (\kappa \nabla u),
\]

with \( \rho \) — is for density, \( e \) — specific (i.e. per unit mass) internal energy, \( \kappa \) — heat conduction coefficient, \( u \) — for temperature.

The simplest case will become the one-dimensional (1D) diffusion equation for \( u \)

\[
\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( K \frac{\partial u}{\partial x} \right),
\]
which is called the equation of \textit{temperature conduction} with \( K = \kappa \cdot \rho^{-1} \left( \frac{d \rho}{dt} \right)^{-1} \), where \( \rho \) is considered constant in space \( \frac{dc}{dt} \) — in space and time.

For uniform mesh, where \( i, i \pm 1 \) denotes \( x, x \pm h \) coordinates, respectively, we have a space discretization

\[
\frac{d}{dx} \left( K \frac{d}{dx} u \right) = \frac{K_{i+1} - u_i - K_{i-1} - u_i}{h^2} = \frac{K_{i+1} - u_i - \left( K_{i+1} + K_{i-1} \right) u_i + K_{i+1} u_i}{h^2}.
\]

In SPH the 1D function \( g(x) \) space-differentiation operator at coordinate \( x_i \) acts in another manner

\[
\frac{d}{dx} g(x) \bigg|_{x_i} = \sum_{j \in \Omega_{w}(h_i)} g(x_j) h_{eff, j} B \frac{d}{dx} w \left( \frac{|x_i - x_j|}{h_i} \right), \quad B = \left( \int_{-\infty}^{\infty} w \left( \frac{|q_i(x)|}{h_i} \right) dq_i \right)^{-1}, \quad q_i(x) = \frac{x_i - x}{h_i},
\]

where \( x_{ij} = x_i - x_j, h_{eff} \) means the \textit{effective} size of “particle” (also thought as discretization point or material point) at coordinate \( x_j \), positive \( h_i \) is the characteristic width of the dimensionless non-negative bell-shaped smoothing function \( w \left( \frac{|x_i - x_j|}{h_i} \right) \) at point \( x_i, \Omega_{w}(h_i) \) is the kernel \textit{support area} (i.e. \( w \left( \frac{|x_i - x_j|}{h_i} \right) \bigg|_{j \in \Omega_{w}(h_i)} = 0 \)), \( B \) is the normalization constant. Kernel \( W(x_{ij}, h_i) = \frac{B}{h_i} w \left( \frac{|x_i - x_j|}{h_i} \right) \) is even function, therefore the support area is presented by the \textit{support radius} (cut-off radius) \( H_{w}(h_i) : j \in \Omega_{w}(h_i) \Rightarrow |x_{ij}| < H_{w}(h_i) \).

Application of space differentiation operators in a sequential manner to obtain the higher-order operator seems bad practice [1, 2], due to necessity to take summation over second stage of neighbourhood (i.e. through neighbours \( k \) of neighbours \( j \) of particle \( i \)), and moreover the additional losses of accuracy on each summation.

Fortunately, some techniques of obtaining higher-order approximation of sequential differentiation operators in SPH were developed [3, 4]. We utilize the universal approach to higher-order approximation of SPH-summation operators representation, where we can chose the desired order of approximation easily, transforming the sequential first-order operators to immediate (pure) versions of higher-order differential operators

\[
\frac{d}{dx} \left( K \frac{d}{dx} u \right) \rightarrow \frac{1}{2} \left( \frac{d^2}{dx^2} (K u) - u \frac{d^2}{dx^2} K + K \frac{d^2}{dx^2} u \right).
\]

As the SPH space-differentiation operator stencil is rather wider, then at typical mesh operator (i.e. \( \{ x + i \cdot h \} \) in 1D, where \( i \in \{-N, \ldots, N\}, N \sim 1 \) in mesh case, and \( \sim 2.5 \div 3 \) — in SPH), the parabolic equations implicit solution methods, widely used in mesh discretizations, seem less efficient applied to SPH. That’s why we try to modify the standard explicit methods for use in SPH approach.

3. Methods developed by Kozyrev&Litvinov (KL) [5]

For linear parabolic equation \( \frac{d}{dt} u(x, t) = \frac{d^2}{dx^2} u(x, t) \) the numerical scheme with two time-layers is

\[
u_{n+1}^{n+1}(x) = G u^n(x), \quad \text{(3)}
\]

where superscripts \( n + 1 \) is for “\( t + \tau \)”, \( n \) — for “\( t \)” time layers, \( G \) is the differential timestep operator.

For \textit{explicit} scheme \( G \rightarrow E + \tau A \),

\[
u_{n+1} = (E + \tau A) u^n, \quad \text{(4)}
\]

2
where \( E \) is the unitary operator, \( A \) — space discretization of parabolic operator \( \frac{\partial^2}{\partial x^2} \):

\[
Eu^n(x) = u^n(x),
\]

\[
Au^n(x) = \frac{u_{i+1}^n - u_i^n - u_i^n - u_{i-1}^n}{h} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}.
\]

Explicit scheme timestep discrete operator \( -\tau A \) is symmetric with non-negative eigenvalues \( \{0 \leq \lambda\} \).

The Fourier transform \( v(\omega) = \int_{-\infty}^{\infty} \exp(-i\omega x)u(x)dx \) gives the

\[
v^{n+1}(\omega) = P_Q(\lambda)v^n(\omega).
\]

\( P_Q(\lambda) \) is the polynomial of the \( Q \)-th order,

\( \lambda(\omega) \) — Fourier representation of the operator \( A \):

\[
\lambda(\omega) = (2 - \exp(i\omega h) - \exp(-i\omega h)) \cdot \frac{\tau}{h^2} = 4C \cdot \sin^2 \left(\frac{h\omega}{2}\right), \quad 0 \leq \omega \leq \frac{\pi}{h},
\]

\( \Rightarrow 0 \leq \lambda \leq \lambda_{\text{max}} = 4C. \)

We chose \( P_Q(\lambda) = 1 - \lambda + \ldots (1 \leq Q) \), which match the \( G = E + \tau A \). This gives

\[
\begin{cases}
P_Q(0) = 1, \\
\frac{dP_Q(\lambda)}{d\lambda}\bigg|_{\lambda=0} = -1.
\end{cases}
\]

For numerical scheme stability \( \max_{0 \leq \lambda \leq 4C} |P_Q(\lambda)| \leq 1 \) is necessary and sufficient.

3.1. based on Chebyshev polynomials

Guess the linear representation \( \lambda = Q^2(1 - \xi) \), \( T_Q(\xi) \) — Chebyshev polynomial. \( \frac{d}{d\xi}T_Q(\xi)\bigg|_{\xi=1} = Q^2. \)

\[
P_Q(\lambda) = T_Q \left( \xi = 1 - \frac{\lambda}{Q^2} \right) = \tilde{T}_Q(\lambda),
\]

\[
\begin{cases}
P_Q(0) = T_Q(1) = 1, \\
\frac{dP_Q(\lambda)}{d\lambda}\bigg|_{\lambda=0} = \frac{dT_Q(\xi)}{d\xi}\bigg|_{\xi=1} = \frac{d\xi}{d\lambda} = Q^2 - \frac{1}{Q^2} = -1.
\end{cases}
\]

\( 0 \leq \lambda \leq 4C \Rightarrow \) for stability \( -1 \leq \xi (\lambda = 4C), \quad -1 \leq 1 - \frac{4C}{Q^2}, \quad 2C \leq Q^2. \)

To write the scheme as predictor–corrector, reform

\[
P_Q(\lambda) = 1 - \lambda \frac{1 - \tilde{T}_Q(\lambda)}{\lambda} = 1 - \lambda \tilde{Z}_{Q-1}(\lambda),
\]

\( \tilde{Z}_{Q-1}(\lambda) = \frac{1 - \tilde{T}_Q(\lambda)}{\lambda} = \frac{1 - T_Q(\xi)}{Q^2(1-\xi)} = Z_{Q-1}(\xi) \) is the Lanczos polynomial.

The iterative

\[
u^n = u^{(0)} \rightarrow \ldots \rightarrow u^{(Q)} = u^{n+1}
\]

numerical scheme from [5] based on Chebyshev polynomials (KL-C):

\[
u^{(1)} = \left( E + \frac{1}{2Q^2} \tau A \right) u^{(0)}, \quad 1 < Q
\]

\[
u^{(\nu)} = \frac{1}{(\nu + 1)^2} \left[ 2\nu^2 \left( E + \frac{1}{Q^2} \tau A \right) u^{(\nu-1)} - (\nu - 1)^2 u^{(\nu-2)} + 2u^{(0)} \right], \quad 1 < \nu < Q
\]

\[
u^{(Q)} = u^{(0)} + \tau A u^{(Q-1)},
\]

\( Q = \left\lfloor \sqrt{2C} \right\rfloor, \) where \( \left\lfloor \cdot \right\rfloor \) denotes the limit to nearest upper integer value.
3.2. based on Lanczos polynomials

Guess the linear representation $\lambda = \frac{Q(Q+2)}{6} (1 - \xi)$, $Z_Q(\xi)$ — Lanczos polynomial.

$$\frac{d}{d\xi} Z_Q(\xi) \bigg|_{\xi=1} = \frac{Q(Q+2)}{6}.$$

$$P_Q(\lambda) = Z_Q \left( \xi = 1 - \frac{6\lambda}{Q(Q+2)} \right) = \tilde{Z}_Q(\lambda),$$

$$\begin{cases}
P_Q(0) = Z_Q(1) = 1, \\
\frac{dP_Q(\lambda)}{d\lambda} \bigg|_{\lambda=0} = \frac{dZ_Q(\xi)}{d\xi} \bigg|_{\xi=1} = \frac{Q(Q+2)}{6} - \frac{6}{Q(Q+2)} = -1.
\end{cases}$$

$0 \leq \lambda \leq 4C \Rightarrow$ for stability $-1 \leq \xi (\lambda = 4C), -1 \leq 1 - \frac{64C}{Q(Q+2)} \leq 12C + 1 \leq (Q + 1)^2$.

To write the scheme as predictor–corrector, reform

$$P_Q(\lambda) = 1 - \lambda = \frac{1 - \tilde{Z}_Q(\lambda)}{\lambda} = 1 - \lambda \tilde{R}_{Q-1}(\lambda),$$

$$\tilde{R}_{Q-1}(\lambda) = 1 - \frac{\tilde{Z}_Q(\lambda)}{\lambda} = \frac{6}{Q(Q+2)} \left( 1 - \xi \right) = R_{Q-1}(\xi), \quad \tilde{R}_{0}(\xi) = 1, \quad R_{1}(\xi) = \frac{\xi+2}{3} = 1 - \frac{2\lambda}{Q(Q+2)},$$

$$R_{\nu+1}(\xi) = \frac{1}{(\nu+2)^2(\nu+3)} \left[ 2\nu(\nu+1)(\nu+2) \xi R_{\nu-1}(\xi) - (\nu-1)\nu^2 R_{\nu-2}(\xi) + 12(\nu+1) \right].$$

Iterative (5) numerical scheme from [5] based on Laczos polynomials (KL-L):

$$u^{(1)} = \left( E + \frac{2}{Q(Q+2)} \tau A \right) u^{(0)}, \quad 1 < Q$$

$$u^{(\nu)} = \left( \frac{1}{(\nu+2)^2(\nu+3)} \left[ 2\nu(\nu+1)(\nu+2) \left( E + \frac{6}{Q(Q+2)} \tau A \right) u^{(\nu-1)} - (\nu-1)\nu^2 u^{(\nu-2)} + 12(\nu+1) u^{(0)} \right] \right), \quad 1 < \nu < Q$$

$$u^{(Q)} = u^{(0)} + \tau A \cdot u^{(Q-1)},$$

$$Q = \left[ \sqrt{12C+1} \right] - 1.$$

(7)

4. “Locally iterative” (LI) numerical schemes [6]

LI and LI-M (i.e. “modified”) schemes for (5):

$$u^{(\nu)} = \frac{1}{1 + a_{\nu}} \left( u^{(0)} + (a_{\nu} E + \tau A) u^{(\nu-1)} \right), \quad 1 \leq \nu \leq Q,$$

$$Q = \begin{cases} p, & \text{LI} \\
2p - 1, & \text{LI-M — only odd integers},
\end{cases}$$

$$p = \left[ \frac{\pi}{4} \sqrt{4C+1} \right] \quad \text{— required order of Chebyshev polynomial},$$

$$a_{\nu} = \frac{4C (\beta_1 - \beta_{\mu(\nu)})}{\beta_1 + 1},$$

$$\beta_\mu = \cos \alpha_\mu, \quad \alpha_\mu = \frac{\theta_\mu p}{2},$$

$$\mu(\nu) = \begin{cases} p + 1 - \nu, & \nu \leq p, \quad \text{LI} \\
p + 1 - \nu, & \nu < p, \quad \text{LI-M \rightarrow} \begin{cases} p, \ldots, 1, \quad \text{LI} \\
p, \ldots, 2p, \ldots, 1, \quad \text{LI-M}.
\end{cases}
\end{cases}$$

$$\mu(Q) = 1 \Rightarrow a_Q = 0, \text{ i.e. the last iteration appears explicit.}$$
To obtain the LI-type scheme of minimal variation, the sequence of Chebyshev polynomial roots should be reordered in a special manner. This ensures stability for non-linear problems.

The \( O(t^2) \) LI-2 scheme can be constructed in a predictor–corrector manner from the LI-M one, we just apply the LI-M for the \( \tilde{\tau} \) as a predictor step, and then perform the corrector step:

\[
\begin{align*}
    u^{(\nu)} &= \frac{1}{1+a_{\nu}} \left(u^{(0)} + \left(a_{\nu}E + \frac{\tau}{2}A\right)u^{(\nu-1)}\right), \quad 1 \leq \nu \leq \tilde{Q}, \\
    \tilde{Q} &= 2\tilde{p} - 1, \\
    \tilde{p} &= \left[\frac{\pi}{4}\sqrt{2C+1}\right], \\
    u^{(Q)} &= u^{(0)} + \tau Au^{(\tilde{Q})}, \\
    Q &= \tilde{Q} + 1 = 2\tilde{p}.
\end{align*}
\]

(9)

5. Runge–Kutta–Legendre (RKL) methods \[7\]
Super-time-stepping methods fall into a category of stabilized Runge–Kutta methods for which additional stages are used to increase the stability of the method, thereby increasing the maximum permissible time-step. This should be contrasted with traditional Runge–Kutta methods, where the additional stages are used to increase the accuracy of the method.

\[
\frac{d}{dt} u = Au \Rightarrow u(t + \tau) = \exp(\tau A)u(t) = (1 + \tau A + \frac{1}{2}(\tau A)^2 + \ldots)u(t).
\]

The internal \( Q \)-stage update of (3) can be expressed in terms of the stability polynomial

\[
u^{n+1}(x) = (E + \tau A)u^n(x) \rightarrow (E + \tau A + \frac{1}{2}(\tau A)^2 + \ldots)u^n(x) = \exp(\tau A)u^n(x) = P_Q(-\tau A)u^n(x),
\]

RKL methods are based on using shifted Legendre polynomials as the stability polynomial of the scheme:

\[
P_Q(\lambda) = a_Q + b_QL_Q(\xi = \psi_0 + \psi_1\lambda) = a_Q + b_Q\tilde{L}_Q(\lambda).
\]

Stability is ensured if \( \forall \lambda : |P_Q(\lambda)| \leq 1 \).

5.1. \( O(t) \) scheme

\[
a_Q = 0, b_Q = 1, \psi_0 = 1, \psi_1 = -\frac{2}{Q(Q+1)} \Rightarrow \xi = 1 - \frac{2}{Q(Q+1)}\lambda, \quad \lambda = \frac{Q(Q+1)}{2}(1 - \xi),
\]

\[
\left\{ \begin{array}{l}
P_Q(0) = L_Q(1) = 1, \\
\frac{dL_Q(\xi)}{d\lambda} \bigg|_{\lambda=0} = \frac{dL_Q(\xi)}{d\xi} \bigg|_{\xi=1} = \frac{Q(Q+1)}{2}, \quad -\frac{2}{Q(Q+1)} = -1.
\end{array} \right.
\]

\[0 \leq \lambda \leq 4C \Rightarrow -1 \leq \xi (\lambda = 4C), \quad -1 \leq 1 - \frac{2}{Q(Q+1)} \cdot 4C, \quad 4C \leq Q(Q+1), \quad Q \geq \frac{1}{2}\sqrt{16C+1} - \frac{1}{2}.\]

We require that every \textit{internal stage} also has a corresponding stability polynomial

\[
u^{(\nu)} = L_{\nu} \left( \xi = 1 - \frac{2}{Q(Q+1)}\right) u^n.
\]

Thus, each stage in the scheme can be thought of as a first-order accurate approximation to the solution at “inner” time \( t = \frac{\nu(Q+1)}{Q(Q+1)}\tau \).

\( O(t) \) Runge–Kutta numerical scheme based on Legendre polynomials (RK1), \( (Q = 1 \rightarrow \text{explicit}) \):

\[
u^{(\nu)} = a_{\nu} \left(E + \frac{2}{Q(Q+1)}\tau A\right)u^{(\nu-1)} + b_{\nu}u^{(\nu-2)}, \quad 1 \leq \nu \leq Q,
\]

\[
a_{\nu} = \frac{2\nu - 1}{\nu} = 1 - b_{\nu},
\]

\[
b_{\nu} = \frac{\nu - 1}{\nu}.
\]

(10)
Figure 1 & Table 1: Confined in \([-1, 1]\) recurrent algebraic polynomials \(f_\nu(\xi)\) from \(\xi \in [-1, 1]\).

| polynomial | equation | \(a\) | \(b\) | \(c\) | \(d\) | \(f_0(\xi)\) | \(f_1(\xi)\) | \(\frac{df_\nu(\xi)}{d\xi}\big|_{\xi=1}\) |
|------------|----------|------|------|------|------|----------|----------|-----------------|
| Chebyshev  | \(T_\nu(\xi) = \cos(\nu \arccos \xi)\) | 1    | 2\xi | -1   | 0    | 1        | 1        | \(\nu^2\)         |
| Lanczos    | \(Z_\nu(\xi) = \frac{1-T_{\nu+1}(\xi)}{(\nu+1)^2(1-\xi)}\) | (\(\nu + 1\))^2 | 2\nu^2\xi | -(\(\nu - 1\))^2 | 2 | 1 | \(\frac{\xi+1}{2}\) | \(\frac{\nu(\nu+2)}{6}\) |
| Legendre   | \(L_\nu(\xi) = \frac{1}{2\nu!} d^{\nu}\left(\xi^2 - 1\right)^\nu\) | \(\nu\) | (2\(\nu - 1\))\(\xi\) | -(\(\nu - 1\)) | 0 | 1 | 1 | \(\frac{\nu(\nu+1)}{2}\) |

5.2. \(O(t^2)\) scheme

\(O(t^2)\) Runge–Kutta numerical scheme based on Legendre polynomials (RKL2),

\(1 < Q:\)

\[
\begin{align*}
\frac{d^n u}{dt^n} &= a_\nu \left( E + \frac{4}{(Q-1)(Q+2)} T A \right) \left( u^{(\nu-1)} - u^{(0)} \right) + b_\nu \left( u^{(\nu-2)} - u^{(0)} \right) \\
&+ \left( E + a_\nu \beta_{\nu-1} \right) \frac{4}{(Q-1)(Q+2)} T A u^{(0)}, \quad 1 \leq \nu \leq Q, \\
\beta_\nu &= \frac{2\nu - 1}{\nu} \frac{\beta_{\nu-1}}{\beta_{\nu-2}}, \\
b_\nu &= -\frac{\nu - 1}{\nu} \frac{\beta_{\nu}}{\beta_{\nu-2}}, \\
a_\nu &= \frac{2\nu - 1}{\nu} \frac{\beta_{\nu}}{\beta_{\nu-1}}, \\
\beta_2 &= \begin{cases} \frac{3}{2} \frac{(\nu-1)(\nu+2)}{2\nu(\nu+1)}, & \nu < 2 \\
\left( \frac{\nu+1}{3} \right), & 2 \leq \nu \end{cases},
\end{align*}
\]

6. Travelling heat wave problem simulation

Simulation setup is: \(x_L \leq x \leq x_R, u(x, 0) = 10^{-10}, K(x, t) = K_0 u^s(x, t), K_0 = 2^s.\)

Boundary conditions are

\[
\begin{cases}
u(x_L, t) = \max(u(x_L, 0), u_{\text{profile}}(x_L, t)), \\
u(x_R, t) = u(x_R, 0).
\end{cases}
\]

\[
u_{\text{profile}}(x, t) = \begin{cases} \left( \frac{2D^s}{k_0}(Dt - x) \right)^{1/s}, & x \leq Dt \\
0, & x > Dt
\end{cases}
\]

stands for the analytical solution of this problem.

where \(D\) treats the heat wavefront propagation velocity.
Table 2: Numerical schemes time-integration characteristics.

| scheme | (eq.) | steps quantity $Q \geq 1$ | steps | time-layers | final iteration | time consistency |
|--------|-------|-----------------|-------|-------------|-----------------|-----------------|
| explicit | (4) | $[2C]$ | 1 | 1 | 1 | 1 | $O(t)$ on iter |
| KL-C | (6) | $\sqrt{2C}$ | 1 | 1 | 3 | +1 | explicit | $O(t)$ final |
| KL-L | (7) | $[\sqrt{12C + 1} - 1]$ | 1 | 2 | 3 | +1 | explicit | $O(t)$ final |
| LI | (8) | $[\frac{1}{3}\sqrt{4C + 1}]$ | 1 | 2 | 2 | +1 | explicit | $O(t)$ final |
| LI-M | (9) | $2[\frac{\pi}{2}\sqrt{4C + 1}] - 1$ | 1 | 3 | 2 | +1 | explicit | $O(t)$ final |
| LI-2 | (9) | $2[\frac{\pi}{2}\sqrt{2C + 1}] \geq 2$ | 2 | 4 | 2 | +1 | explicit | $O(t^2)$ final |
| RKL1 | (10) | $[\frac{1}{2}(\sqrt{16C + 1} - 1)]$ | 1 | 1 | 3 | +0 | non-explicit | $O(t)$ every |
| RKL2 | (11) | $[\frac{1}{2}(\sqrt{32C + 9} - 1)] \geq 2$ | 2 | 2 | 3 | +1 | non-explicit | $O(t^2)$ every |

Figure 2: *Single* steps quantity during overall integrated time vs. parabolic Courant number for such time.

We chose $x_L = 0$, $x_R = 1$, $D = 4$ and run simulations till $t_{end} = 0.2$ in two cases: with $s = 4$ and $s = 8$. Timestep $\tau$ was defined from the relation $\tau_D = \alpha$, where $h = 10^{-2}$ is the mesh resolution. And we vary the amount of mesh cells the thermal wave propagates per $\tau$ choosing $\alpha$ as 0.5 or 1.

The obtained results of simulations are presented on figures 3–4 (horizontal axis — $x$, vertical

Table 3: $u(x, t_{end})$ relative $L_2$ error, %

| $\alpha$ | scheme | $K \sim u^4 / u^8$ not refreshed | $K \sim u^4 / u^8$ refreshed |
|-----------|--------|---------------------------------|-----------------------------|
| 1 | KL-L | 4.69 / 2.28 | 0.67 / 3.01 |
| 1 | LI-M | 4.71 / 7.5 | 0.64 / 3.2 |
| 0.5 | KL-L | 2.31 / 0.67 | 2.03 / 5.4 |
| 0.5 | LI-M | 2.38 / 0.69 | 2.02 / 5.39 |
7. Summary

At table 1 the formulation and properties of three (Chebysev, Lanczos, Legendre) confined in unity algebraic polynomials \(|f_\nu(\xi)| \leq 1\) are given, on figure 1 the first 6 degrees \((\nu = 0, \ldots, 5)\) of each polynomial are plotted with horizontal axis for \(\xi\), vertical — for \(f_\nu(\xi)\). The \(-1 \leftarrow \xi\) part of the polynomial argument values causes the growth of the numerical noise during the simulation. So in this case behaviour of the Lanczos polynomial is optimal.

The required total number of numerical steps during the same simulation period \(T\) is represented on figure 2. On the horizontal axis is the Courant parameter \(C = \frac{TK}{h^2}\) for the parabolic diffusion equation \(\frac{\partial}{\partial t}u = K \frac{\partial^2}{\partial x^2}u\). Explicit scheme needs the \(2C\) explicit steps permanently, therefore the benefit of iterative schemes with required steps number \(\sim \sqrt{C}\) will increase with the rise of \(T\). Important properties of all schemes are displayed on table 2, they

---

---
are — the overall steps quantity $Q$, required simultaneous time-layers number, and the final iteration kind.

The KL-L and LI-M schemes are preferable due to best stability properties, curiously their lowest performance reasoned by the highest $Q$ against rest iterative methods. The LI-schemes need to store less time-layers simultaneously. So the most recommended is the slightly less performant KL-L scheme owing to stability based on its polynomial, while LI-M is slightly ($\sim 10\%$) more performant. This schemes afford the ability to iterate $K(u)$ along with scheme iterations. Such practice has no strict proof, as the iterations number $Q$ is fixed by initial $K(u^{(0)})$ while the real $K(u^{(Q)})$ is changing. But it obtains the smoother resulting profiles, remaining within the heat wave velocity timestep limitation $\frac{\kappa D}{h} \leq \sim 1$.

Regarding to (1), it can be solved in the following way: find $u^{(Q-1)}$ using iterative scheme for (2), and then relay on

$$e^{(Q)} = e^{(0)} + \frac{1}{\rho} \tau \tilde{A} u^{(Q-1)},$$

where $\nabla (\kappa \nabla u) \rightarrow \tilde{A}$. RKL-schemes have a drawback, being non-explicit on the last iteration, becoming prohibited in this approach, opposite the advantage is their consistency after every iteration, not only the whole timestep passed, combined with approximately twice greater productivity towards KL-L.

The investigated iterative methods has demonstrated their productivity for solving the heat conduction equation applied to SPH, giving the desired simulation speedup. Which makes them promising at high-performance massively-parallel calculations according to their explicit nature.

The detailed comparison with implicit methods on stiff 2D and 3D simulations especially should be held in advance.

References

[1] Brookshaw L 1994 Mem. S.Alt. 65 1033–1042
[2] Cleary P W and Monaghan J J 1999 Journal of Computational Physics 148 227–264
[3] Jubelgas M, Springel V and Dolag K 2004 Monthly Notices of the Royal Astronomical Society 351 423–435
[4] Stranex T and Wheaton S 2011 Computer methods in applied mechanics and engineering 200 392–402
[5] Kozyrev O M and Litvinov V P 2012 Affairs of Nuclear Science and Engineering (VANT). Series: Matem. Model. Phys. Proc. (4) 3–12 in russian
[6] Zhukov V T 2011 Mathematical Models and Computer Simulations 3 311–332
[7] Meyer C D, Balsara D S and Aslam T D 2014 Journal of Computational Physics 257 594–626