Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 956–969
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fast method for solving the heat equation by layer potentials

Johannes Tausch *

Southern Methodist University, Department of Mathematics, 209A Clements Hall, Dallas, TX 75275-0156, United States

Received 21 April 2006; received in revised form 18 October 2006; accepted 1 November 2006
Available online 19 December 2006

Abstract

Boundary integral formulations of the heat equation involve time convolutions in addition to surface potentials. If M is the number of time steps and N is the number of degrees of freedom of the spatial discretization then the direct computation of a heat potential involves order N^2M^2 operations. This article describes a fast method to compute three-dimensional heat potentials which is based on Chebyshev interpolation of the heat kernel in both space and time. The computational complexity is order p^4q^2NM operations, where p and q are the orders of the polynomial approximation in space and time.

Keywords: Heat equation; Layer potentials; Fast multipole method; Fast Gauss transform

1. Introduction

Boundary integral formulations of parabolic problems involve time convolutions in addition to the usual boundary integral operators. While the finite element or finite difference method are highly popular for this type of problems, the method of layer potentials has distinct advantages, in particular, stability of explicit time stepping and reduction of unknowns to the boundary surface. Boundary element methods that evaluate the time convolution have been discussed in the past decades in several places, such as [4,9,11].

If the time convolution is computed directly the cost quickly becomes prohibitively expensive with increasing problem size which makes the integral equation approach less attractive than competing approaches. However, the situation changes if fast methods for the computation of heat potentials are considered. One such an algorithm is the fast Gauss transform [7]. In this approach the heat kernel is approximated by a Hermite expansion, which leads to an efficient scheme to compute the spatial convolution efficiently. This approach has been combined with re-starts to avoid time convolutions [12]. The original version of the fast Gauss Transform is based on Hermite expansions of the heat kernel. Since the spatial variables of the kernel separate, the translation operator appear in tensor product form, which can be exploited to reduce the

* Tel.: +1 214 768 2515; fax: +1 214 768 2355.
E-mail address: tausch@mail.smu.edu.

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.11.001

computational cost associated with translation operators [13]. Besides Hermite expansions, exponential expansions have been considered as well, in this case translation operators are diagonal [8].

Another technique that exploits the convolution form of the time integral is to evaluate heat potentials in Fourier domain. This was described for bounded domains in [6], where the Green's function is expanded in a Fourier series. The method must be combined with nonuniform FFTs since the heat sources are located on a boundary surface. In unbounded domains the kernel appears as a continuous Fourier transform which results in some nontrivial complications, cf. [5].

The approach described in this article is more close to the fast Gauss Transform in that the computation is performed in 'physical' and not in Fourier domain. Hence, the method is somewhat simpler when the solution must be evaluated in physical space in every time step, which is the case, for instance, when the heat equation is solved via the Green's representation theorem. The main features of the discussed approach are as follows:

- The heat kernel is expanded in both space and time. The time convolution is computed using a hierarchical scheme, similar to the FMM for one-dimensional source distributions. However, certain modifications are necessary to account for the Volterra form of heat potentials. The cost per time step in this algorithm is almost independent of the length of the time interval.
- The heat kernel is interpolated at the Chebyshev nodes in space and time. The translation operators of the multipole method can be easily derived and preserve the tensor product form of the kernel.

After discussing heat potentials and their discretization in Section 2 we introduce the Chebyshev interpolation and the resulting translation operators in Sections 3 and 4. This leads to the fast algorithm described in Section 5. Section 6 concludes with numerical results.

2. Heat potentials

It is well known that the solution of the heat equation satisfies Green's Representation formula. If $u_t = \Delta u$ in the exterior of a domain with boundary S and if u has vanishing initial conditions, then Green's formula assumes the form

$$\frac{1}{2}u(x,t) = \mathscr{K}u(x,t) - \mathscr{V}\frac{\partial u}{\partial n}(x,t), \quad x \in S, t > 0. \tag{1}$$

In three spatial variables, the single- and double layer potential are given by

$$\mathscr{V}g_s(x,t) = \int_0^t \int_S G(x-y,t-\tau)g_s(y,\tau)\,\mathrm{d}S_y\,\mathrm{d}\tau,$$
$$\mathscr{K}g_d(x,t) = \int_0^t \int_S \frac{\partial}{\partial n_y} G(x-y,t-\tau)g_d(y,\tau)\,\mathrm{d}S_y\,\mathrm{d}\tau,$$

where S is the boundary surface, and G is the heat kernel which is given by

$$G(r,t) = \frac{1}{(4\pi t)}^{3/2} \exp\left(-\frac{|r|^2}{4t}\right).$$

For the following discussion it is convenient to write the single layer potential in the form

$$\mathscr{V}g_s(x,t) = \frac{1}{\sqrt{4\pi}}\int_0^t \frac{1}{\sqrt{t-\tau}} V(t-\tau)g_s(\tau)\,\mathrm{d}\tau, \tag{2}$$

where

$$V(\delta)g_s(x,\tau) = \frac{1}{4\pi\delta}\exp\left(-\frac{|x-y|^2}{4\delta}\right)g_s(y,\tau)\,\mathrm{d}S_y, \tag{3}$$

and the double layer potential in an analogous manner.

The kernel of (3) is the Green's function of the two-dimensional heat equation. Using local parameterizations, it can be shown that this operator is q-times differentiable in δ for every point $x \in S$ if the surface is of class C^{q+1}. The first term of the Taylor series is given by

$$V(\delta)g_s(x) = g_s(x) + \mathrm{O}(\delta), \tag{4}$$

$$K(\delta)g_d(x) = H(x)g_d(x) + \mathrm{O}(\delta), \tag{5}$$

where $H(x)$ is the mean curvature of the surface.

The direct boundary element method consists of discretizing (1) and solving for the missing boundary data. The usual discretization schemes for integral operators, namely the Galerkin, collocation and the Nyström method are available. In this work we consider the latter, because no influence coefficients must be computed or stored, which constitutes a significant savings over the Galerkin or collocation method.

2.1. Temporal discretization

The Nyström method replaces integrals by quadrature rules and enforces the equation on the quadrature nodes. For the temporal discretization we consider the composite midpoint rule of meshwidth Δt and quadrature nodes $t_j = (j + 1/2)\Delta t$. Because of the $\mathrm{O}((t-\tau)^{-\frac{1}{2}})$-singularity of the integrand in (2) this rule is applied to the interval $[0, i\Delta t]$, where the integrand is smooth. For the singular piece $[i\Delta t, t_i]$ expansion is used. The resulting rule is

$$\frac{1}{\sqrt{4\pi}}\int_0^{t_i}\frac{V(t_i-\tau)}{\sqrt{t-\tau}}g_s(\tau)\,\mathrm{d}\tau \approx \frac{\Delta t}{\sqrt{4\pi}}\sum_{j=0}^{i-1}\frac{V(t_i-t_j)}{\sqrt{t_i-t_j}}g_s(t_j) + \sqrt{\frac{\Delta t}{2\pi}}g_s(t_i).$$

A straightforward analysis shows that the error of the singular part is $\mathrm{O}(\Delta t^{3/2})$ whereas the error of the midpoint rule is $\mathrm{O}(\Delta t^{1/2})$. To improve this convergence rate, we introduce the following singularity subtraction

$$\frac{1}{\sqrt{4\pi}}\int_0^{t}\frac{V(t-\tau)}{\sqrt{t-\tau}}g_s(\tau)\,\mathrm{d}\tau = \frac{1}{\sqrt{4\pi}}\int_0^{t}\frac{1}{\sqrt{t-\tau}}(V(t-\tau)g_s(\tau) - V(0)g_s(t))\,\mathrm{d}\tau + \sqrt{\frac{t}{\pi}}g_s(t),$$

where the last term comes from expansion (4). Now the integrand is $\mathrm{O}((t-\tau)^{\frac{1}{2}})$ which implies that the error of the composite midpoint rule is improved to $\mathrm{O}(\Delta t^{3/2})$. The desingularized midpoint rule simplifies to

$$\frac{1}{\sqrt{4\pi}}\int_0^{t_i}\frac{V(t_i-\tau)}{\sqrt{t-\tau}}g_s(\tau)\,\mathrm{d}\tau \approx \frac{\Delta t}{\sqrt{4\pi}}\sum_{j=0}^{i-1}\frac{V(t_i-t_j)}{\sqrt{t_i-t_j}}g_s(t_j) + m_i g_s(t_i), \tag{6}$$

where

$$m_i = \sqrt{\frac{t_i}{\pi}} - \frac{\Delta t}{\sqrt{4\pi}}\sum_{j=0}^{i-1}\frac{1}{\sqrt{t_i-t_j}}.$$

The analogous rule for the double layer potential is

$$\frac{1}{\sqrt{4\pi}}\int_0^{t_i}\frac{K(t_i-\tau)}{\sqrt{t-\tau}}g_d(\tau)\,\mathrm{d}\tau \approx \frac{\Delta t}{\sqrt{4\pi}}\sum_{j=0}^{i-1}\frac{K(t_i-t_j)}{\sqrt{t_i-t_j}}g_d(t_j) + m_i H g_d(t_i), \tag{7}$$

where H denotes multiplication with the mean curvature. The desingularized rules (6) and (7) are $\mathrm{O}(\Delta t^{3/2})$ and differ from the regular rule only by the weight of the node t_i.

2.2. Spatial discretization

The surface integral operators in the above quadrature rules have smooth kernels, thus the Nyström method can also be used for the spatial discretization. Quadrature rules for surface integrals are usually based on a triangulation of the surface and piecewise polynomial interpolation. Their form is

$$\int_S f(x)\,\mathrm{d}s(x) \approx \sum_P f(x_P) w_P. \tag{8}$$

We use the quadratic rule where the quadrature points x_P are located on the midpoints of the edges of the triangles; the details of the derivation of this rule can be found in Atkinson [1] and Chien [3]. Since the kernels in (6) become increasingly peaked when Δt is decreased, the meshwidth Δx of the surface triangulation must be decreased as well. The form of the heat kernel suggests that the relationship between the two mesh parameters should be at least $\Delta x \sim (\Delta t)^{\frac{1}{2}}$.

2.3. Nyström method

We now turn to the discretization of (1). The surface integral operators in the desingularized rules (6) and (7) are replaced by the spatial rule (8) for time steps $j = 0, \ldots, i-2$. The term $j = i-1$ is replaced by the truncated local expansions (4) and (5), which introduces an additional error of $\mathrm{O}(\Delta t^{3/2})$. The resulting recurrence is

$$\frac{1}{2} u_P^i = m_i H_p u_P^i - m_i v_P^i + \sqrt{\Delta t}\left(H_p u_P^{i-1} - v_P^{i-1}\right) - \Phi_P^i, \tag{9}$$

where u_P^i is the approximation for $u(t_i, x_P)$ and v_P^i is the approximation for $\frac{\partial}{\partial n} u(t_i, x_P)$ and

$$\Phi_P^i = \Delta t \sum_{j=0}^{i-2} \sum_Q w_Q \left(G(t_i - t_j, x_P - x_Q) v_Q^j - \frac{\partial}{\partial n_y} G(t_i - t_j, x_P - x_Q) u_Q^j \right). \tag{10}$$

The smooth potential Φ_P^i contains only quantities of the previous time steps, hence (9) can be solved for the unknown boundary data. By far the dominant cost in every time step the computation of Φ_P^i. In the following we describe a fast method for this task.

3. Chebyshev expansion of the heat kernel

Chebyshev polynomials are commonly used to approximate functions in the interval $[-1, 1]$. In this article, we use the definition

$$T_n(x) = c_n \cos(n \arccos(x)), \quad c_n = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0, \\ 1, & n > 0. \end{cases}$$

Note that the inclusion of the factor c_n is not standard, but will simplify the formulas below. The roots of T_{n+1} are given by

$$\omega_k^n = \cos\left(\frac{\pi}{2} \frac{2k+1}{n+1}\right), \quad 0 \leqslant k \leqslant n$$

and the Lagrange polynomials with respect to these nodes are

$$L_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{x - \omega_k^n}{\omega_i^n - \omega_k^n}.$$

The interpolation $\mathscr{T}_n f$ of a function f at the Chebyshev nodes can be expressed in terms of Lagrange polynomials

$$\mathscr{T}_n f(x) = \sum_{i=0}^{n} f(\omega_i^n) L_i(x) \tag{11}$$

as well as in terms of Chebyshev polynomials

$$\mathscr{T}_n f(x) = \sum_{\alpha=0}^{n} \hat{f}_\alpha T_\alpha(x), \tag{12}$$

where

$$\hat{f}_\alpha = \frac{2}{n+1} \sum_{k=0}^{n} T_\alpha(\omega_k^n) f(\omega_k^n). \tag{13}$$

The last formula follows from the discrete orthogonality of the Chebyshev polynomials, see, e.g. [2].

The essence of most fast methods for integral operators is to replace the kernel by a truncated series expansion. Since (11) and (12) represent the same polynomial they can be used alternatively. In order to preserve the tensor product form of the translation operators discussed below, we will use the Lagrange form for the temporal interpolation and the Chebyshev expansion for the spatial interpolation.

Let v be a cube in $\mathbb{R}^3$ with center $\bar{x}$ and side length $2h_x$ and $\tilde{v}$ be another cube with same side length and center $\tilde{x}$, furthermore let I and $\tilde{I}$ be non overlapping intervals of $\mathbb{R}$ with length $2h_t$ and centers $\bar{t}$ and $\tilde{t}$, respectively. For $x \in v$, $y \in \tilde{v}$, $t \in I$ and $\tau \in \tilde{I}$ we introduce local variables with the transformation

$$\begin{aligned} &x = \bar{x} + x' h_x, \quad y = \tilde{x} + y' h_x, \quad x', y' \in [-1,1]^3 \\ &t = \bar{t} + t' h_t, \quad \tau = \tilde{t} + \tau' h_t, \quad t', \tau' \in [-1,1]. \end{aligned}$$

When the heat kernel is considered as a function of the eight local variables, the Chebyshev interpolation using order p polynomials in space and order q polynomials in time is

$$\begin{aligned} \frac{1}{(4\pi)^{3/2}} \frac{1}{(t-\tau)^{3/2}} \exp\left(-\frac{|x-y|^2}{4(t-\tau)}\right) &= \frac{1}{(4\pi h_t)^{3/2}} \frac{1}{(d+t-\tau')^{3/2}} \exp\left(-\rho \frac{|r+x'-y'|^2}{d+t'-\tau'}\right) \\ &\approx \frac{1}{(4\pi h_t)^{3/2}} \sum_{\substack{0 \leqslant i \leqslant q \\ 0 \leqslant j \leqslant q}} \sum_{|\alpha+\beta| \leqslant p} E_{i\alpha,j\beta} L_i(t') L_j(\tau') T_\alpha(x') T_\beta(y'). \end{aligned} \tag{14}$$

Here, α and β are multiindices, $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, $|\alpha| = \alpha_1 + \alpha_2 + \alpha_3$, and $T_\alpha(x') = T_{\alpha_1}(x'_1) T_{\alpha_2}(x'_2) T_{\alpha_3}(x'_3)$. The constants $d > 0$, $r \in \mathbb{R}^3$ and $\rho > 0$ are given by

$$d = (\bar{t} - \tilde{t})/h_t, \tag{15}$$

$$r = (\bar{x} - \tilde{x})/h_x, \tag{16}$$

$$\rho = \frac{h_x^2}{4h_t}. \tag{17}$$

It follows from (11) and (13) that the coefficients in (14) are given by

$$E_{i\alpha,j\beta} = E^{(1)}_{i\alpha_1,j\beta_1} E^{(2)}_{i\alpha_2,j\beta_2} E^{(3)}_{i\alpha_3,j\beta_3}, \tag{18}$$

where

$$E^{(\sigma)}_{i\gamma,j\delta} = \left(\frac{2}{p+1}\right)^2 \frac{1}{\left(d + \omega_i^q - \omega_j^q\right)^{\frac{1}{2}}} \sum_{\substack{0 \leqslant k \leqslant p \\ 0 \leqslant l \leqslant p}} T_\gamma(\omega_k^p) T_\delta(\omega_l^p) \exp\left(-\rho \frac{(r_\sigma + \omega_k^p - \omega_l^p)^2}{d + \omega_i^q - \omega_j^q}\right).$$

3.1. Approximation errors

Since the heat kernel is an analytic function in all local variables the approximation error of the Chebyshev exhibits exponential decay up to logarithmic factors [10]. To illustrate this behavior, we compute the Chebyshev interpolation error of the function

$$G(r,s) = \frac{1}{(d+s)^{1/2}} \exp\left(-\rho \frac{r^2}{d+s}\right), \quad |s|, |r| \leqslant 1.$$

Fig. 1 shows the maximum interpolation error as a function of p and q when $d = 4$ and $\rho = 10$. Since the figure is on a logarithmic scale, the exponential decay in the s-variable appears as a straight line. In the r-variable G is

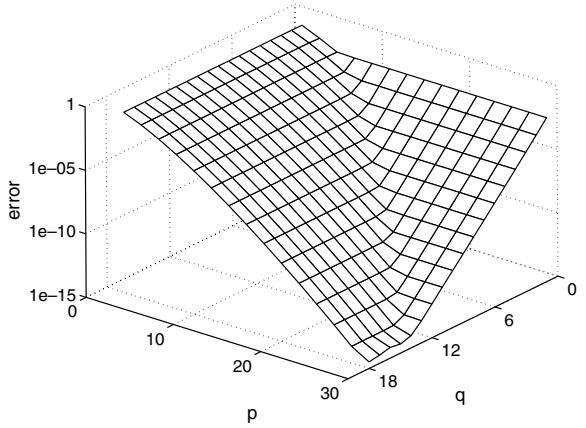

Fig. 1. Maximum errors of the Chebyshev interpolation as a function of p and q; $\rho = 10$ and $d = 4$.

an entire function, therefore the decay of the error is initially faster than exponential in p until the error of the s-variable is dominant.

4. Translation operators

The FMM combines sources and evaluation points in a tree of cubes. To accelerate the computation of cube interactions the kernel is replaced by a truncated series expansion. The efficient evaluation of such a series involves the moments of the source cube and the expansion coefficients of the destination cube. In the course of the computation, the moments and coefficients must be translated between the different levels of the cube hierarchy.

In addition to the space variable, heat potentials involve an integration over the time variable. Therefore, the time interval $[0, T]$, in which the heat equation is to be solved, is also clustered into a binary tree of intervals. The finest level, denoted by level zero, consists of M intervals of length Δt. The half-length of the finest level intervals is also denoted by $h_{t0} = \Delta t/2$. The half length of a level-l interval is $h_t = 2^l h_{t0}$. Likewise, the half-length of a cube in the finest spatial level is h_{x0} and the half-length in level l is $h_x = 2^l h_{x0}$.

We briefly describe the translation operators related to Chebyshev interpolation in space and time.

4.1. Moments-to-local (MtL) translations

The moments-to-local (MtL) translation is the evaluation of a heat potential in $I \times v$ due to sources in $\tilde{I} \times \tilde{v}$ using the expansion (14). The potential can be any combination of single- and double-layer sources. Replacing the kernel by the Chebyshev interpolation results in the series

$$\sum_{\substack{t_Q \in \tilde{I},\\ x_P \in S_{\tilde{v}}}} \left(G(x - x_P, t - t_Q) g_s(x_P, t_Q) + \frac{\partial}{\partial n_y} G(x - x_P, t - t_Q) g_d(x_P, t_Q) \right) w_P \Delta t$$

$$\approx \frac{1}{(4\pi h_t)^{3/2}} \sum_{0 \leqslant i \leqslant q} \sum_{|\alpha| \leqslant p} \lambda_{i\alpha} L_i(t') T_\alpha(x').$$

The summation in the left-hand side is over quadrature points in $\tilde{I} \times S_{\tilde{v}}$ and primes denote local variables. The expansion coefficients $\lambda_{i\alpha}$ are given by

$$\lambda_{i\alpha} = \sum_{j=0}^{q} \sum_{|\beta| \leqslant p - |\alpha|} E_{i\alpha, j\beta} \mu_{j\beta}. \tag{19}$$

The $\mu_{j\beta}$'s are the moments of the sources in $\tilde{I} \times \tilde{v}$, given by

$$\mu_{j\beta} = \sum_{\substack{t_Q \in \tilde{I}, \\ x_P \in S_{\tilde{v}}}} L_j(t'_Q) \left(T_\beta(x'_P) g_s(x_P, t_Q) + \frac{\partial}{\partial n_y} T_\beta(x'_P) g_d(x_P, t_Q) \right) w_P \Delta t.$$

Because the expansion coefficients in (18) appear as products, the MtL transform (19) is a tensor product in the space variables, which can be exploited to reduce the cost of computing all $\lambda_{i\alpha}$, $i = 0, \ldots, q$, $|\alpha| \leqslant p$ to $O(q^2 p^4)$ operations.

4.2. Moments-to-moments (MtM) translations

In the FMM sources and evaluation points are combined in a tree of cubes in space and a tree of time intervals. The moments are computed in a recursive fashion, beginning with the finest level. This involves recentering moments from a child $\tilde{v} \times \tilde{I}$ to the parent $v \times I$, this operation is also known as the MtM translation and is described next.

The MtM translation operators follows from the following addition formula for the Chebyshev polynomials:

$$T_\gamma\left(\frac{1}{2}x \pm \frac{1}{2}\right) = \sum_{\delta=0}^{\gamma} a_{\gamma\delta} T_\delta(x), \tag{20}$$

where

$$a_{\gamma\delta} = \frac{2}{p+1} \sum_{k=0}^{p} T_\delta(\omega_k^p) T_\gamma\left(\frac{1}{2}\omega_k^p \pm \frac{1}{2}\right).$$

This formula directly follows from (13) and holds when $p \geqslant \gamma$. A similar result holds for the Lagrange polynomials

$$L_i\left(\frac{1}{2}t \pm \frac{1}{2}\right) = \sum_{j=0}^{q} b_{ij} L_j(t), \tag{21}$$

where

$$b_{ij} = L_i\left(\frac{1}{2}\omega_j^q \pm \frac{1}{2}\right).$$

Let $(\bar{x}, \bar{t})$ be the center and $(4h_x, 4h_t)$ be the lengths of the parent $v \times I$. The lengths of the child $\tilde{v} \times \tilde{I}$ are $(2h_x, 2h_t)$ and the centers are $\tilde{x} = \bar{x} + \sigma_x h_x$ and $\tilde{t} = \bar{t} + \sigma_t h_t$ where $\sigma_t, \sigma_{x,k} \in \{\pm 1\}$. The contribution $\mu_{i\alpha}$ of the child's moments $\tilde{\mu}_{j\beta}$ to the parent's moment is

$$\mu_{i\alpha} = \sum_{\substack{t_Q \in \tilde{I}, \\ x_P \in S_{\tilde{v}}}} L_i\left(\frac{t_Q - \bar{t}}{2h_t}\right) \left(T_\alpha\left(\frac{x_P - \bar{x}}{2h_x}\right) g_s(x_P, t_Q) + \frac{\partial}{\partial n_x} T_\alpha\left(\frac{x_P - \bar{x}}{2h_x}\right) g_d(x_P, t_Q) \right) w_P \Delta t.$$

Changing the local variables with respect to $v \times I$ to variables local to $\tilde{v} \times \tilde{I}$ gives

$$\frac{t - \bar{t}}{2h_t} = \frac{1}{2}\frac{t - \tilde{t}}{h_t} + \frac{1}{2}\sigma_t,$$
$$\frac{x - \bar{x}}{2h_x} = \frac{1}{2}\frac{x - \tilde{x}}{h_x} + \frac{1}{2}\sigma_x.$$

Combining this with (20) and (21) results in

$$\mu_{i\alpha} = \sum_{j=0}^{q} b_{ij} \sum_{\beta \leqslant \alpha} a_{\alpha_1\beta_1} a_{\alpha_2\beta_2} a_{\alpha_3\beta_3} \tilde{\mu}_{j\beta}.$$

This translation is again of tensor-product form in the spatial variable. Thus the complexity of one MtM transform is $O(q^2p^4)$.

In the algorithm described in the following there are also MtM translations from one temporal interval to a subinterval within the same cube. In this case the translation formula involves only in the spatial variable

$$\mu_{i\alpha} = \sum_{j=0}^{q} b_{ij}\tilde{\mu}_{j\alpha}.$$

This MtM translation has $O(q^2p^3)$ complexity.

4.3. Local-to-local (LtL) translations

In the LtL translation the Chebyshev expansion for $v \times I$ is recentered to the child $\tilde{v} \times \tilde{I}$.

Let $(\bar{x}, \bar{t})$ be the center and $(4h_x, 4h_t)$ be the lengths of the parent $v \times I$. The lengths of the child $\tilde{v} \times \tilde{I}$ are $(2h_x, 2h_t)$ and the centers are $\tilde{x} = \bar{x} + \sigma_x h_x$ and $\tilde{t} = \bar{t} + \sigma_t h_t$ where σ_t, $\sigma_{x,k} \in \{\pm 1\}$. The translation from the parent's expansion to the child's expansion follows from the addition formulas (20) and (21):

$$\begin{aligned}\Phi(x,t) &= \sum_{\substack{|\alpha|<p,\\ 0\leqslant i\leqslant q}} \lambda_{\alpha i} L_i\left(\frac{t-\bar{t}}{2h_t}\right) T_\alpha\left(\frac{x-\bar{x}}{2h_x}\right) = \sum_{\substack{|\alpha|<p,\\ 0\leqslant i\leqslant q}} \lambda_{\alpha i} L_i\left(\frac{1}{2}\frac{t-\tilde{t}}{h_t} - \frac{\sigma_t}{2}\right) T_\alpha\left(\frac{1}{2}\frac{x-\tilde{x}}{h_x} - \frac{\sigma_x}{2}\right)\\ &= \sum_{\substack{|\alpha|<p,\\ 0\leqslant i\leqslant q}} \tilde{\lambda}_{\alpha i} L_i\left(\frac{t-\tilde{t}}{h_t}\right) T_\alpha\left(\frac{x-\tilde{x}}{h_x}\right),\end{aligned}$$

where

$$\tilde{\lambda}_{\alpha i} = \sum_{j=0}^{q} b_{ji} \sum_{\substack{\beta \geqslant \alpha\\ \beta \leqslant p}} a_{\alpha_1\beta_1} a_{\alpha_2\beta_2} a_{\alpha_3\beta_3} \lambda_{\beta j} \tag{22}$$

are the expansion coefficients for $\tilde{v} \times \tilde{I}$. The tensor product form in the above formula can be used to compute an LtL translation in $O(q^2p^4)$ operations.

If the translation is only in time (i.e., $v = \tilde{v}$), then (22) simplifies to

$$\tilde{\lambda}_{\alpha i} = \sum_{j=0}^{q} b_{ji} \lambda_{\alpha j}$$

which can be computed in $O(q^2p^3)$ operations.

5. A fast algorithm for the time convolution

This section describes in detail the fast method for the smooth potential (10).

5.1. Temporal tree

Recall that the half-lengths of the intervals in the finest level of the temporal tree are $h_{t,0} = \Delta t/2$, hence the ith interval I_i^0 is $[2i\, h_{t,0}, 2(i+1)h_{t,0}]$ and its center is the quadrature node t_i. The time quadrature of the smooth potential (10) is over intervals $I_0^0 \cup \ldots \cup I_{i-2}^0$. That is, there is one interval between the integration interval and the interval where the smooth potential is evaluated.

The finest-level intervals are the leafs of the binary tree, cf. Fig. 2. Let the binary representation of i be

$$i = \sigma_p 2^p + \cdots + \sigma_1 2 + \sigma_0, \tag{23}$$

then the position of I_i^0 in the tree can be found as follows. Start in the leftmost node of the $p+1$st level, then for $l = p, \ldots, 0$ follow the left branch when $\sigma_l = 0$ and the right branch when $\sigma_l = 1$. The *parent* of I_i^0 in the lth level is denoted by $I_{i_l}^l$ where i_l is given by

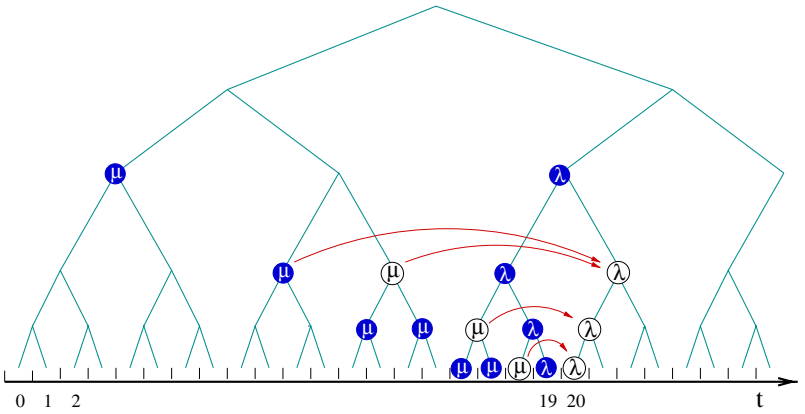

Fig. 2. Time step for $i = 20$, μ's represent moments, and λ's represent expansion coefficients. Quantities on dark ground have been computed in the previous time steps, quantities on white ground are computed in this time step. Curved arrows indicate MtL translations.

$$i_l = \sigma_p 2^{p-l} + \cdots + \sigma_{l+1} 2 + \sigma_l. \tag{24}$$

The interval $I^l_{i_l}$ has length $h_{t,l} = 2^l h_{t,0}$ and is given by

$$I^l_{i_l} = \bigcup_{j=2^l i_l}^{2^l(i_l+1)-1} I^0_j.$$

The *neighbors* of $I^l_{i_l}$ are $\mathcal{N}^l_i = \{I^l_{i_l-1}, I^l_{i_l}\}$ for $i_l > 0$ and $\mathcal{N}^l_0 = \{I^l_0\}$ for $i_l = 0$. Because of the Volterra form of the integral operators this definition only involves intervals corresponding to earlier times. The *interaction list* $\mathcal{I}^l_{i_l}$ of $I^l_{i_l}$ consists of nodes in level l, which are children of $I^l_{i_l}$'s parent's neighbors, but not neighbors of $I^l_{i_l}$. Because of the simple structure of the binary tree, there are only two nontrivial cases of interaction lists. If the binary representation of $i > 1$ is given by (23), then

$$\mathcal{I}^l_{i_l} = \begin{cases} \{I^l_{i_l-2}\}, & \text{if } \sigma_l = 0, \\ \{I^l_{i_l-3}, I^l_{i_l-2}\}, & \text{if } \sigma_l = 1. \end{cases}$$

For $i = 0$ or $i = 1$ the interaction lists $\mathcal{I}^l_{i_l}$ are empty. Thus for $i_l > 1$ there are either one or two intervals in an interaction list, depending on whether the node is the right or the left branch of its parent. If $t \in I^0_i$ then the interval of integration in (10) is the union of the interaction lists of all parents of I^0_i, that is,

$$[0, 2(i-1)h_{t0}] = \bigcup_{l=0}^{p} \mathcal{I}^l_{i_l}. \tag{25}$$

5.2. Spatial tree

Similar to the FMM in potential theory, we introduce a hierarchical decomposition of $\mathbb{R}^3$ into cubes. In the finest level the cubes have side length $2h_{x,0}$, where $h_{x,0}$ is a parameter at our disposition, whose choice will be discussed later. The next coarser level consists of cubes of side length $4h_{x,0}$ which contain eight finest level cubes. The coarsest level cube has side length $2^{L_s+1}h_{x,0}$ and contains the surface S. The set of nonzero cubes in level l_S is denoted by C_{l_S}.

The heat kernel is smooth for $t > \tau$ and exponentially decaying in space, therefore it suffices to compute MtL translation between neighboring cubes only. When the separation of t and τ is small, the heat kernel is peaked and must be resolved by fine-level cubes and fine-level intervals. For larger separations of t and τ the kernel can be resolved in coarse-level cubes and intervals. The number of neighbors in MtL translations is independent of the level.

To control the errors introduced by the Chebyshev expansion and by the far field truncation the relationship between the temporal level l and the spatial level l_S is such that the parameter ρ in the Chebyshev expansion of the heat kernel (14) remains bounded. Because of (17) the dependence of the temporal level l to the spatial level l_S is

$$l_S = \min\left(\left[\frac{l}{2}\right], L_S\right), \tag{26}$$

where $[\cdot]$ denotes rounding off to the next smaller integer. Thus for time-levels $l \leqslant 2L_S$ the values of ρ are either ρ_0 or $\rho_0/2$, where

$$\rho_0 = \frac{h_{x0}^2}{4h_{t0}}. \tag{27}$$

5.3. The fast algorithm

To compute the right-hand side in (10) efficiently the heat potentials are expanded in local expansions that are valid in a given interval of the time-tree and in a given cube of the corresponding level of the spatial tree.

Specifically, in the ith time step the vector λ_v^l is the vector of expansion coefficients that is valid in $I_{i_l} \times S_v$ representing heat sources in $\mathscr{I}_{i_l}^l \times S$. Here, i and i_l are given by (23) and (24), respectively.

Because of (25) the smooth potential can be computed by translating and adding expansions from the coarser to the finer levels and then evaluating the resulting expansion in the finest level.

The λ_v^l's are computed using the MtL translations, the moments are computed using the MtM translations.

The smooth potential in the ith time step depends only on the λ_v^l's in I_i^0's parents $I_{i_l}^l$, $l = 0, \ldots, p$. Hence, only one set of expansion coefficients per temporal level must be stored and updated. Since the expansion coefficients in the coarser levels are valid in larger time intervals it is not necessary to update all coefficients in all levels in every time step. Indeed, if $\sigma_0 = \ldots \sigma_{L-1} = 0$ in (23) then only the λ_v^l's in levels $0 \leqslant l \leqslant L$ must be updated.

The expansion coefficients of $I_{i_l}^l$ depend on the moments of $I_{i_l-2}^l$ and $I_{i_l-3}^l$, denoted by $\mu_{2,v}^l$ and $\mu_{3,v}^l$. Thus, it is possible to store and update only two sets of expansion coefficients in each level.

One time step of this algorithm is illustrated in Fig. 2. A more detailed description follows.

MtL translations for levels $0 \leqslant l \leqslant L$

```
l = L
for = v ∈ C_{l_S}
   for = v' ∈ N(v)
      form λ_v^l by translating μ_{2v'}^l and μ_{3v'}^l
   end
end
for l = L − 1, ..., 0
   for = v ∈ C_{l_S}
      for = v' ∈ N(v)
         form λ_v^l by translating μ_{2v'}^l
      end
   end
end
```

MtM translations for levels $0 \leqslant l \leqslant L$

```
l = L
for = v ∈ C_{l_S}
   overwrite μ_{3v}^l with μ_{2v}^l
end
```

for $l = L, \ldots, 1$
  for $= v \in C_{l_S}$
    if $= l$ is even and $[l/2] \leqslant L_S$
      $\mathscr{C}(v)$:=Children of v
    else
      $\mathscr{C}(v)$:=$\{v\}$
    end if
    for $= v' \in \mathscr{C}(v)$
      form μ_{2v}^{l} by translating $\mu_{2v'}^{l-1}$ and $\mu_{3v'}^{l-1}$
    end
  end
end

LtL translations for levels $0 \leqslant l \leqslant L$

for $l = L + 1, \ldots, 1$
  for $= v \in C_{l_S}$
    if $= l$ is even and $[l/2] \leqslant L_S$
      $\mathscr{C}(v)$:=Children of v
    else
      $\mathscr{C}(v)$:=$\{v\}$
    end if
    for $= v' \in \mathscr{C}(v)$
      translate λ_{v}^{l} and add to $\lambda_{v'}^{l-1}$
    end
  end
end

Algorithm for computing the smooth potential

for $i = 2, \ldots, M$
  Determine p and L from the binary expansion of i.
  Perform MtM translations
  for $= v \in C_0$
    Compute the moments $\mu_{2,v}^{0}$
  end
  Perform MtL translations
  Perform LtL translations
  for $v \in C_0$
    Evaluate the expansions using the coefficients λ_{v}^{0}
  end
end

5.4. Complexity

The binary tree has $O(M)$ nodes. Each node is made up of cubes in the corresponding spatial level. Using geometric series arguments it follows that the number of all nonempty cubes in all nodes is $O(NM)$. For every cube one set of moments and one set of expansion coefficients is computed using a bounded number of translation operators. The estimates in Section 4 imply that the total complexity is $O(p^4 q^2 NM)$.

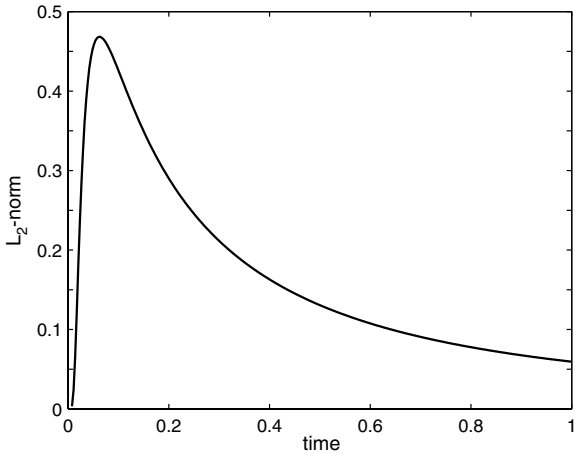

Fig. 3. L_2-norm of the solution of the sample problem as a function of time.

6. Numerical example

To illustrate the convergence behavior we solve the heat equation in the exterior of the unit sphere. The solution approach is based on the Green's Representation formula (1), desingularized Nyström discretization, and the fast method to evaluate the heat potentials. To obtain a problem with known solution we place the heat source

$$u(x,t) = \frac{1}{(4\pi t)^{3/2}} \exp\left(-\frac{|x-x_0|^2}{4t}\right) \tag{28}$$

at the point $x_0 = [0.5, 0, 0]$, supply the Neumann data on the sphere and compare the numerical solution on the sphere with the exact solution (28). The L_2 norm of the solution as a function of time is shown in Fig. 3

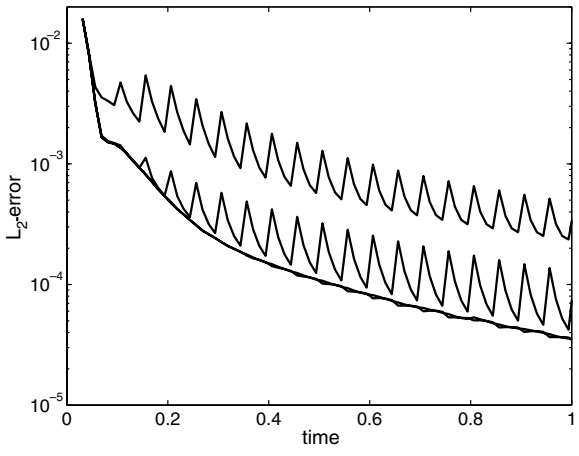

Fig. 4. L_2-Norm of the error for $p = 12$ (top), $p = 16$, $p = 20$, and $p = 24$ (bottom). The last two lines overlap.

Table 1
Discretization parameters for results in Table 2 and Figs. 5 and 6

Mesh	# Time steps	# Triangles	p	q	L_S	ρ_0
1	20	192	12	4	1	2.5
2	40	768	16	4	1	5
3	80	3072	20	4	1	10
4	160	12,288	24	4	2	5
5	320	12,288	28	6	2	10

Table 2
Results for the meshes in Table 1

Mesh	cpu Time	Ratio	Error	Rate
1	4	3.3e − 9	3.85e − 3	
2	20	6.4e − 10	7.42e − 4	2.37
3	131	2.1e − 10	1.02e − 4	2.86
4	7218	7.0e − 10	3.48e − 5	1.55
5	45844	5.3e − 10	1.24e − 5	1.49

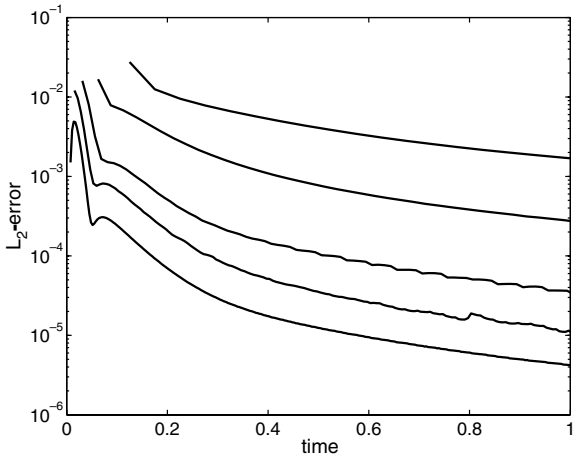

Fig. 5. L_2-errors corresponding to discretization meshes in Table 1; coarsest (top) to finest (bottom) mesh.

6.1. Choice of parameters

In the discretization scheme considered, the time step size Δt has the greatest influence on the error. This parameter determines the quadrature error of the temporal midpoint rule as well as the truncation error in the expansion of the local part.

The parameter of greatest influence to the error of the fast method is ρ_0, because it controls the error due to neglecting well-separated interactions. Increasing ρ_0 reduces this error but demands for larger values of p and q. Once Δt and ρ_0 are determined, the cube width h_{x0} follows from (27). Table 2 and Figs. 5 and 6 show the L_2-error for $t = 0.5$ and cpu timings for refining the meshwidth. The table also displays the ratio of the cpu time to the complexity estimate p^4q^2NM and the convergence rate of the error.

Fig. 4 displays the L_2-errors of the solution of the test problem as a function of time for increasing values of p. In this case, there are 80 time steps and 3072 triangles.

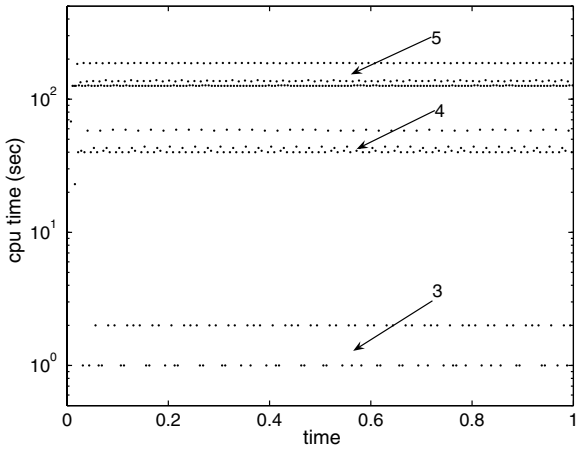

Fig. 6. Cpu times (in s) per time step as a function of time. The numbers refer to the meshes of Table 1. The cpu times for Meshes 1 and 2 resulted are too short to be measured accurately and are not shown.

7. Conclusion

We have discussed a fast method for solving heat conduction problems with layer potentials. The numerical results are in good agreement with the $O(p^4q^2NM)$ complexity and the $O(\Delta^{3/2})$ quadrature error. It is possible to construct higher-order quadrature schemes that evaluate more terms of the expansion of the local part. The difficulty is that the higher terms involve derivatives of the surface curvature and get very complicated.

For the simplicity of exposition, we have restricted the discussion to the homogeneous heat equation and vanishing initial conditions. Source terms lead to volume potentials which can be treated with the same interpolation scheme of the kernel. Non-trivial initial conditions can be included with well-documented algorithms, such as the fast Gauss transform.

References

[1] Kendall E. Atkinson, The Numerical Solution of Integral Equations of the Second Kind, Cambridge University Press, 1997.
[2] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zhang, Spectral Methods in Fluid Dynamics, Springer, 1988.
[3] D. Chien, Numerical evaluation of surface integrals in three dimensions, Math. Comp. 64 (1995) 727–743.
[4] G.F. Dargush, P.K. Banerjee, Application of the boundary element method to transient heat conduction, Int. J. Numer. Meth. Eng. 31 (1991) 1231–1247.
[5] L. Greengard, P. Lin, Spectral approximation of the free-space heat kernel, Appl. Comput. Harmonic Anal. 9 (1999) 83–97.
[6] L. Greengard, J. Strain, A fast algorithm for the evaluation of heat potentials, Comm. Pure Appl. Math. XLIII (1990) 949–963.
[7] L. Greengard, J. Strain, The fast Gauss transform, SIAM J. Sci. Statist. Comput. 12 (1991) 79–94.
[8] L. Greengard, X. Sun, A new version of the fast Gauss transform, Doc. Math. J. DMV Extra Volume ICM 1998 III (1998) 575–584.
[9] M.M. Grigoriev, G.F. Dargush, Higher-order boundary element methods for transient diffusion problems. part I: bounded flux formulation, Int. J. Numer. Meth. Eng. 55 (2002) 1–40.
[10] J. Mason, D. Handscomb, Chebyshev Polynomials, CRC Press, 2002.
[11] E.A. McIntyre, Boundary integral solutions of the heat equation, Math. Comp. 46 (173) (1986) 71–79.
[12] J. Strain, Fast adaptive methods for the free-space heat equation, SIAM J. Sci. Statist. Comput. 15 (1) (1994) 185–206.
[13] X. Sun, Y. Bao, A Kronecker product representation of the fast Gauss transform, SIAM J. Matrix Anal. Appl. 24 (3) (2003) 768–786.