CTDS – Mauling Heat Equation on Unit Disk by Conformal Parametrization

Martin Holzinger

Institute of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10,
1040 Vienna, Austria; *martin.holzinger@tuwien.ac.at*

SNE 30(2), 2020, 43-50, DOI: 10.11128/sne.30.tn.10511
Received: March 15, 2020; Revised April 24, 2020
Accepted: April 30, 2020
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Let us consider a heat conduction problem on the unit square, solve it analytically and compare this series solution with both the results obtained by MATLAB/PDE-Toolbox (using the FE-method) and a self-implementation achieved with *Mathematica* (using the method of lines). Extending Finite-Difference formulae to higher precision gives rise to the idea of utilizing the CTDS-method, best suitable on regular and equidistant grids, also on other domains. By introducing apt coordinates one is therefore able to do a parametrization, e.g. of the unit disk, by the square. Conformal transformation from square to disk provides this parametrization, the original implementation can easily be extended and we find by again comparing a series solution to our obtained simulation results, that order of convergence is being preserved. Moreover, our conformal transformation provides the fundamental tensor and no further structural errors are being introduced as the involved elliptic functions can be evaluated to arbitrary precision.

Introduction

By application of the CTDS-method ("Continuous Time Discrete Space") spatial derivatives in a PDE are replaced by their Finite-Difference approximations which yields a coupled system of ODEs getting stiffer with grid refinement. These systems can be treated with standard algorithms and we show that order of convergence can easily be adjusted to the requirements. Restricted to the unit square and regular grids, conformal maps then provide access to more general domains.

1 Series Solution on Unit Square

For a given heat conduction problem on unit square $\mathfrak{Q}$:

$$u_t = \kappa \triangle u, \quad u(x,y,0) = 1$$

with NEUMANN boundary conditions

$$\nabla_{\mathbf{n}} u(1,y,t) = \nabla_{\mathbf{n}} u(-1,y,t) = \nabla_{\mathbf{n}} u(x,1,t) = 0$$

and a NEWTON-type BC on the south side,

$$\nabla_{\mathbf{n}} u(x,-1,t) + \gamma u(x,-1,t) = 0,$$

a series solution can be obtained [4] by means of eigenvalues and eigenfunctions of the negative Laplacian:

$$u(x,y,t) = \sum_{j=1}^{\infty} \frac{2}{\xi_j} \frac{\sin \xi_j}{1 + \frac{2\gamma}{4\gamma^2 + \xi_j^2}} \cos \frac{\xi_j (y-1)}{2} e^{-\frac{\kappa \xi_j^2}{4} t},$$

where (for $\gamma > 0$) the ξ_j are solutions to $\tan \xi = 2\gamma/\xi$. This rapidly converging series ($t > 0$) is well suited for a comparison with simulation solutions on an $n \times n$-grid

$$\{(-1 + \frac{2(i-1)}{n-1}, -1 + \frac{2(j-1)}{n-1}); i,j = 1, \dots n\}.$$

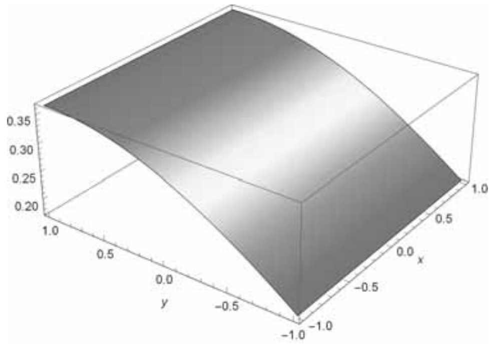


Figure 1: Analytical solution on $\mathfrak{Q}$ ($t = 4, \gamma = \kappa = 1$).

```
n=101; nx=ny=n; dx=2/(nx-1); dy=2/(ny-1); dx2=dx^2; dy2=dy^2; tend=5;
Z=Array[z,{nx,ny}]; X=Array[zreal,{nx,ny}]; Y=Array[zimag,{nx,ny}];
For[i=1,i≤nx,i++,For[j=1,j≤ny,j++,Z[[i,j]]=-1+(i-1)dx+I(-1+(j-1)dy);];]

func[t_][i_,j_]:=u[i,j][t]; U=Array[func[t],{nx,ny}]; X=Re[Z]; Y=Im[Z];

u[1,j_][t_]:=u[2,j][t]; u[nx,j_][t_]:=u[nx-1,j][t];
u[i_,1][t_]:=u[i,2][t]/(1+dy); u[i_,ny][t_]:=u[i,ny-1][t];

uxx[i_,j_][t_]:=(u[i+1,j][t]-2u[i,j][t]+u[i-1,j][t])/dx2;
uyy[i_,j_][t_]:=(u[i,j+1][t]-2u[i,j][t]+u[i,j-1][t])/dy2;

eqn=Table[u[i,j]'[t]==uxx[i,j][t]+uyy[i,j][t],{i,2,nx-1},{j,2,ny-1}];
ic=Table[u[i,j][0]==1,{i,2,nx-1},{j,2,ny-1}];eqns=Flatten[Join[eqn,ic]];
vars=Flatten[Table[u[i,j][t],{i,2,(nx-1)},{j,2,(ny-1)}]];

result=NDSolve[eqns,vars,{t,0,tend},InterpolationPrecision→20,
      Compiled→True,Method→{"EquationSimplification"→"Residual"}];

U=U/.result; lsg=Flatten[Table[{X[[i,j]],Y[[i,j]],
              SetAccuracy[U[[1,i,j]]/.  t→4,50]},{i,1,n},{j,1,n}],1];
```

2 Simulation Solution on Square

On a computational grid like above, functions $u_{ij}(t)$ are now taking over and are being considered in each grid point (x_i,y_j). Replacing the Laplacian and the boundary conditions with their discrete approximations yields a consistent system of ODEs to the original problem. The boxed Mathematica code demonstrates a crude but fully functional implementation that we shall call *FD1*. Note that two-point forward/backward boundary approximations and a five-star for the Laplacian in the interior points only produce a first-order convergent algorithm.

Finite difference formulae. Let us therefore first look out for higher order finite-difference approximations to the spacial derivatives involved. SCHIESSER[6] provides some of them and according to FORNBERG[1], his method is smartly being generalized to our two-dimensional needs: Suppose we are interested in finding the approximation

$$\frac{\partial^{\mu+\nu}u(x,y)}{\partial^{\mu}x\partial^{\nu}y} \approx \sum_{m=-m_0}^{m_1-m_0}\sum_{n=-n_0}^{n_1-n_0} a_{mn}u(x+m\Delta_x, y+n\Delta_y)$$

for a (potentially mixed) partial derivative of orders μ and ν in x- and y-direction. For generally given gridpoints let now Mathematica develop

$$\frac{\xi^{m_0}\eta^{n_0}(\ln\xi)^{\mu}(\ln\eta)^{\nu}}{\Delta_x{}^{\mu}\Delta_y{}^{\nu}} \approx \sum_{m=0}^{m_1}\sum_{n=0}^{n_1} a_{m-m_0,n-n_0}\xi^m\eta^n$$

in the Taylor series at $(\xi=1,\eta=1)$ to find the weights a_{mn} for the approximation in a specific gridpoint. Equidistance between the gridpoints is supposed but with m_0,m_1,n_0,n_1 we are able to claim how many neighbour points to the left/right and above/below should be incorporated which is necessary to specifically handle boundary or boundary-near points, depending on the desired order of convergence (expand to the proper Taylor polynomial). That is, m_0 is the count of intervals between the utmost left point and the point of interest whereas m_1 counts the intervals between the utmost points and similar for n_0,n_1 in y-direction.

For example, if we are interested in a fourth-order approximation of the x- derivative in a gridpoint lying on the left side of our unit square, we set $\mu=1$, $\nu=0$, $m_1=4$ and $m_0=n_0=n_1=0$ to find

$$\frac{\ln\xi}{\Delta_x} \approx = \frac{1}{12\Delta_x}\left(-25+48\xi-36\xi^2+16\xi^3-3\xi^4\right)$$

which means that at the left boundary we can change the first-order approximations in our code to

$$u' \approx (-3u_5+16u_4-36u_3+48u_2-25u_1)/12\Delta_x.$$

The other boundary sides are treated very similar and with $u' = 0$ or $u' = \gamma u_1$ (take in the latter case care of the derivative orientation) we solve this for u_1 to obtain a fourth-order accurate boundary approximation. It is clear that to obtain a global fourth order convergent algorithm, interior points for the Laplacian have to be treated in an appropriate way. Taking this into consideration, for $i = 3, \dots n-2$ we make use of

$$u_{xx}(x_i, y_j, t) \approx \frac{-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} - u_{i-2}}{12\Delta^2},$$

whereas in case of $i = 2$ (cases $i = n-1$ and u_{yy} analog)

$$u_{xx}(x_i, y_j, t) \approx \frac{u_6 - 6u_5 + 14u_4 - 4u_3 - 15u_2 + 10u_1}{12\Delta^2}$$

can be deployed. Replace the blue/bf highlighted code and you are ready to run a higher order code, *FD2*.

Results. With the presented series solution, we are able to validate the implementation of our CTDS-Code and also compare it against the results obtained by MATLAB's PDE-Toolbox. Let $\hat{u}(x,y,t)$ be the series solution on $\mathfrak{Q}$, then by considering the relative error (cf. Figure 2, Figure 3) in a grid point (x_i, y_j),

$$e_{i,j}(t) := \frac{\hat{u}(x_i, y_j, t) - u(x_i, y_j, t)}{\hat{u}(x_i, y_j, t)},$$

one has a tool for such a comparison at hand. Another possible way for checking the results is regarding the absolute errors $E_{i,j}(t) := |u(x_i, y_j, t) - \hat{u}(x_i, y_j, t)|$, see Figure 4, or their arithmetic average,

$$E(t) := \frac{1}{n^2} \sum_{i,j} E_{i,j}(t).$$

Using different sized grid spacings on $\mathfrak{Q}$ (e.g. 51×51, 101×101 and 201×201 is a good idea) by halving Δ also gives a clue about orders of convergence. $\mathcal{O}(\Delta)$ and $\mathcal{O}(\Delta^4)$ are as expected for *FD1* and *FD2* while *PdeTbx* shows up quadratically, cf. Table 1.

As a conclusion, major benefits in applying CTDS-methods are that orders of convergence can easily be adjusted and higher order difference formulae show up with formidable error behaviour even on coarse grids which helps to save computational time. Furthermore, replacing spatial derivatives by discrete differences gives rise to treatment of more general classes of PDEs while $PdeTbx$ has its limitations in this regard.

A severe limitation is that our method is bound to orthogonal grids (and working best on equidistant grid sizes). We focus on that by regarding conformal maps.

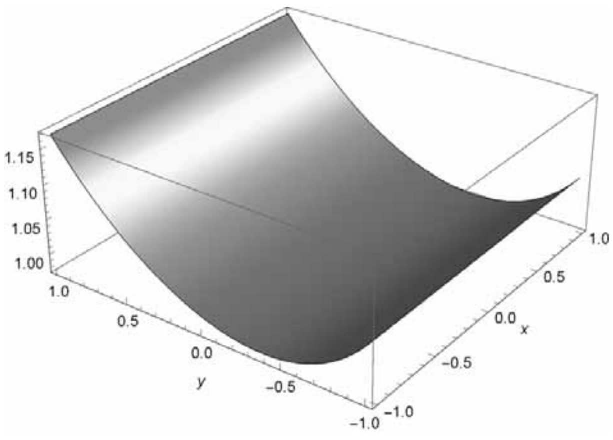


Figure 2: Rel. errors $e_{i,j}(4)$ on $\mathfrak{Q}$ for $n = 101$, method $FD1$.

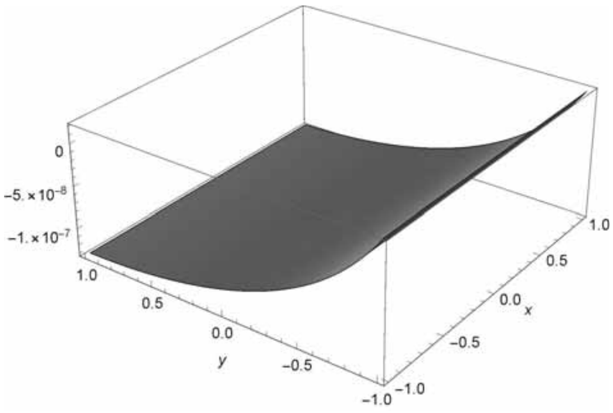
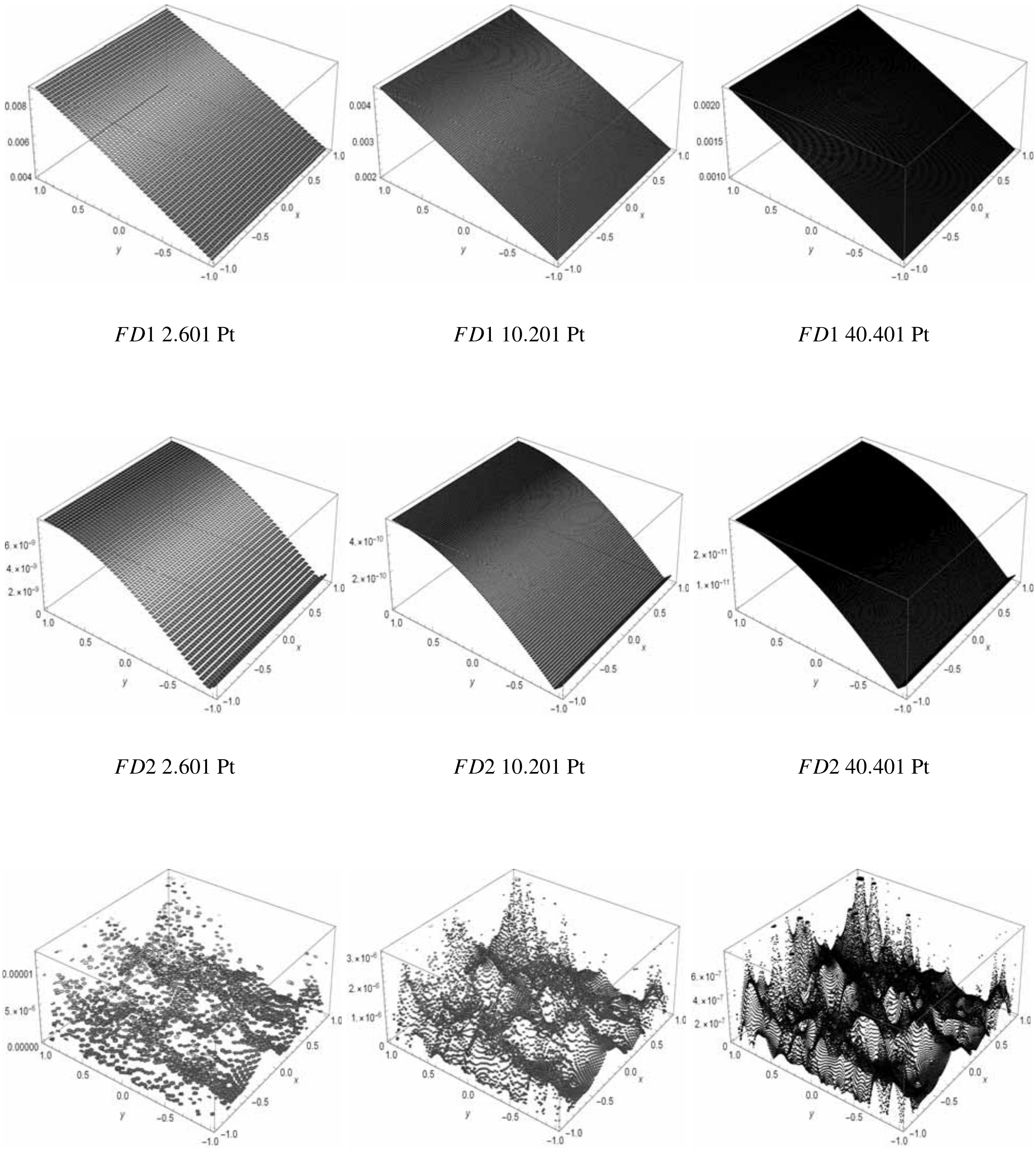


Figure 3: Rel. errors $e_{i,j}(4)$ on $\mathfrak{Q}$ for $n = 101$, method $FD2$.

Method	Points	Avg. Abs. Error	Calc. Order
$FD1$	2.601	0.00627	—
$FD1$	10.201	0.00312	2.00717
$FD1$	40.401	0.00156	2.00360
$FD2$	2.601	$4.67382 * 10^{-9}$	—
$FD2$	10.201	$2.99177 * 10^{-10}$	15.62228
$FD2$	40.401	$1.82074 * 10^{-11}$	16.43158
$PdeTbx$	2.577	$3.47640 * 10^{-6}$	—
$PdeTbx$	10.145	$8.64349 * 10^{-7}$	4.02199
$PdeTbx$	40.257	$2.15899 * 10^{-7}$	4.00349

Table 1: Comparing mean arithmetic absolute errors, $t = 4$.



$FD1$ 2.601 Pt

$FD1$ 10.201 Pt

$FD1$ 40.401 Pt

$FD2$ 2.601 Pt

$FD2$ 10.201 Pt

$FD2$ 40.401 Pt

$PdeTbx$ 2.577 Pt

$PdeTbx$ 10.145 Pt

$PdeTbx$ 40.257 Pt

Figure 4: Absolute errors $E_{i,j}(4)$ for different grid sizes on $\mathfrak{Q}$.

3 Conformal Transformations

Conformal maps preserve angles between intersecting grid lines. Moreover, using cartesian co-ordinates on $\mathfrak{Q}$, local base vectors in a specific grid point are rotated but stretched by the same factor by such a transformation, $f(z)$. To construct the map itself, we argument by means of a *point transformation* of the complex plane while mapping the square to a given domain (demonstrated with the unit disk, $\mathfrak{C}$) will later on be interpreted as a *transformation of co-ordinates* to simulate a PDE on this domain. Under this point of view the given domain is being parameterized by the unit square.

Regarding our heat equation, the Laplacian is not invariant but transforms in a very convenient way, merely a (from point to point differing) factor $1/|f'(z)|^2$ appears to correct the impact of the new curvilinear but orthogonal co-ordinates. Keeping this in mind, we compute the map from $\mathfrak{Q}$ to $\mathfrak{C}$, save its derivative values in each gridpoint and use them in a simulation study on $\mathfrak{C}$.

Back in about 1869 H.A. SCHWARZ and E.B. CHRISTOFFEL independently found that the upper complex plane $\Im(w) > 0$ (or $\mathfrak{C}$ respectively, just consider a Moebius transformation) can be conformally mapped to a polygonal arc with vertices z_k by elliptic integrals [2]

$$z = f(w) = C + D\int_0^w \prod_{k=1}^{n}(\zeta - w_k)^{-\beta_k}\mathrm{d}\zeta.$$

The constants C and D act as translation and rotation/stretching respectively, to map $\mathfrak{C}$ to $\mathfrak{Q}$ we can fix them immediately demanding $f(0) = 0$ and $f(1) = 1$. With $\pi\beta_k$ denoting the outer tangent turning angle in vertex z_k, we determine $\beta_k = 1/2$ for all four square vertices and the vertex preimages w_k can in this case be prescribed by $w_{k+1} = \exp[i(\frac{\pi}{4} + \frac{k\pi}{2})]$, $k = 0,\ldots 3$. Note that for more general polygonal arcs finding the preimages can be a tough task [7]. Putting all togheter,

$$z = f(w) = -\frac{2\sqrt[4]{-1}}{\mathfrak{C}_{k;\frac{1}{2}}}\,\mathfrak{C}_f\left[i\,\mathrm{arsinh}\left(\sqrt[4]{-1}\,w\right);-1\right]$$

maps the disk to our unit square where elliptic functions help us to express the map in closed form. As f is bi-holomorphic and we are interested in the inverse transformation from square to disk, Mathematica handles this by the definition -(-1)$^{1/4}$JacobiSN[z*EllipticF[I*ArcSinh[(-1)$^{1/4}$],-1],-1]. Obviously not only the conformal map but also its derivatives can be evaluated to any desired precision.

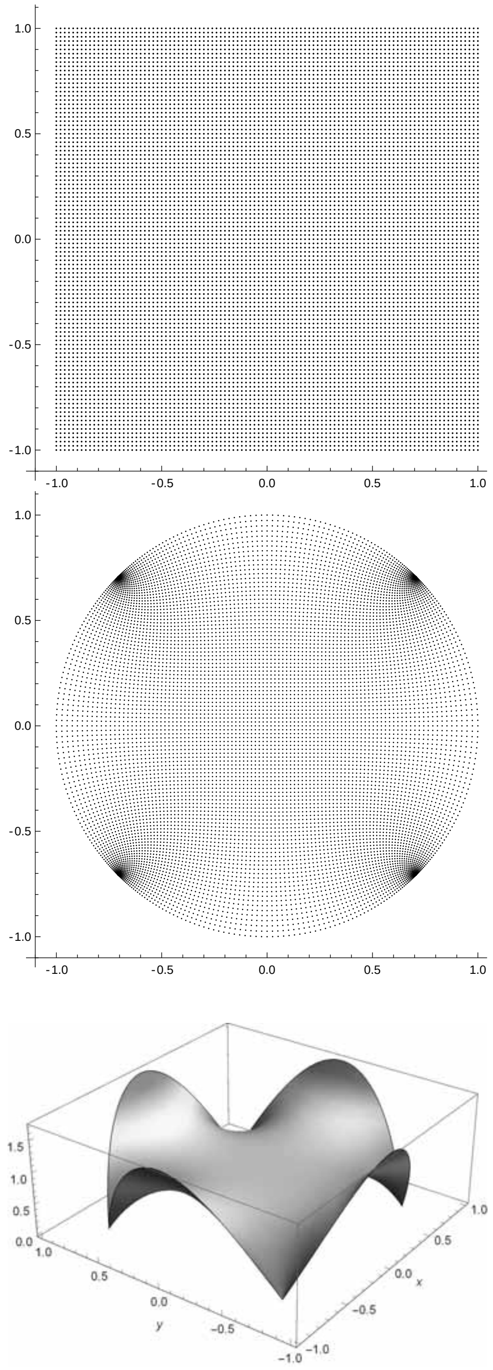


Figure 5: Computational grid ($n = 101$) before (t) and after (c) transformation with $|f'(x,y)|^2$ shown at the bottom. The conformal map from square to disk can also be interpreted as an introduction of curvilinear co-ordinates on $\mathfrak{C}$ which changes the fundamental tensor from $\delta_{ij} \to |f'(z)|^2\delta_{ij}$.

4 Series Solution on Unit Disk

Let us now consider a heat conduction problem on $\mathfrak{E}$:

$$u_t = \kappa \triangle u, \quad u(\xi,\eta,0) = 1 - \left(\xi^2 + \eta^2\right)$$

with DIRICHLET boundary condition

$$u(\xi,\eta,t) = 0, \quad |\zeta| = 1 \quad \forall \zeta = \xi + i\eta.$$

Similar to the problem on $\mathfrak{Q}$ a (symmetric) solution on $\mathfrak{E}$ can be derived by means of BESSEL functions [3]

$$u(r,t) = 4\sum_{k=1}^{\infty} \frac{e^{-\kappa t \lambda_{0,k}^2}\, \mathfrak{J}_0(r\lambda_{0,k})\, \mathfrak{J}_2(\lambda_{0,k})}{\lambda_{0,k}^2\left(\mathfrak{J}_0(\lambda_{0,k})^2 + \mathfrak{J}_1(\lambda_{0,k})^2\right)}$$

with zeroes $\lambda_{0,k}$ of $\mathfrak{J}_0(x)$ and (for t>0) staggering convergence behavior, thus well suited for comparison with simulation results (Figure 6). If you do not like to calculate by hand, in case of our specific problem you can even let Mathematica do the job[1] for you.

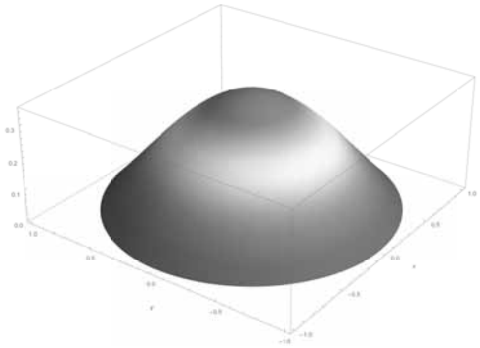


Figure 6: Analytical solution on $\mathfrak{E}$ ($t = 2, \kappa = 1/10$).

5 Simulation Solution on Disk

To get the simulation of the disk problem ready to run, one merely has to numerically calculate the first derivatives $f'(z) = \xi_x + i\eta_x$ of the conformal transformation on any considered grid to the desired precision, save them as a list on the grid and merely replace the Laplacian on the righthandside of the equation:

$$\triangle u \to \frac{\triangle u}{\xi_x^2 + \eta_x^2}.$$

Obviously our boxed Mathematica code is very easy to adapt, we just have to modify the bf/italic expression. Note that the singularities arising in the vertex points caused by the map can be neglected since values at these points are not being used in the simulation run and must not be provided. This argument is given with respect to transformed Newton boundary conditions.

As for the pre-given Dirichlet boundary conditions, it turns out that convergence order of $FD1$ increases to be quadratic, by replacing the first section of the blue-colored code dealing with the boundary conditions by assignment of fixed zero values our simulation code is ready to run and comparable to the series solution.

For a valid comparison with $PdeTbx$ results, it is now necessary to focus on grid sizes with a similar amount of grid points and for that reason the grids $46 \times 46, 91 \times 91$ and 181×181 seem to be appropriate.

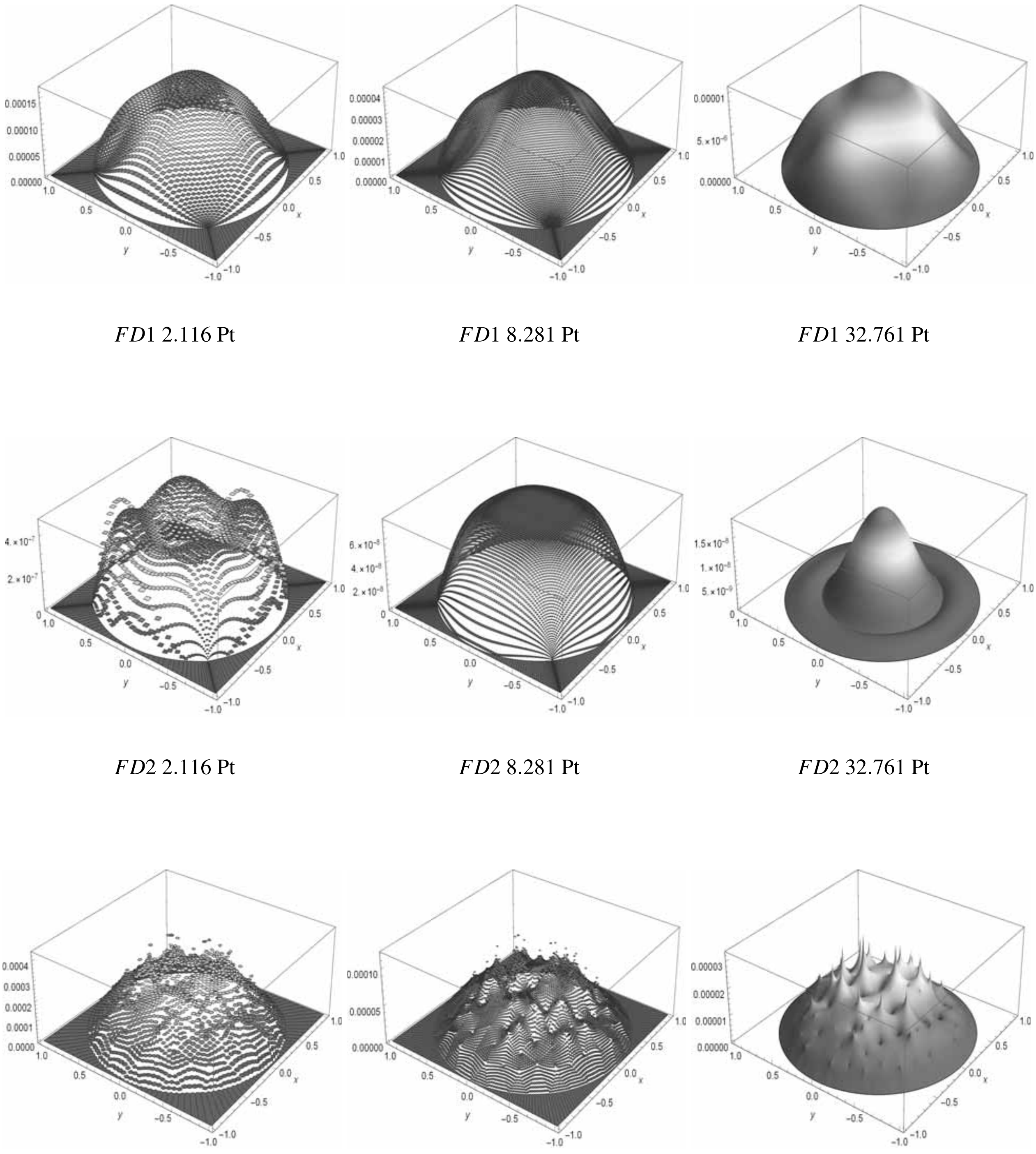
Table 2 shows that even algorithm $FD1$ performs slightly better than $PdeTbx$ with respect to averaged absolute errors. As for $FD2$, note that forth order convergence is not achieved when switching from $46 \times 46 \to 91 \times 91$ grid but only shows up with a factor 5.7 instead of about 16 - we suspect this is essentially caused by the fact that in this case of an even number of gridpoints the origin point is not being used in the calculations. Overall absolute errors on disk are shown in Figure 7.

For our specific heat conduction problem on unit disk it enlightens that application of conformal transformation had no impact of error or convergence behavior (numerical values calculated up to 50 digits precision).

Method	Points	Avg. Abs. Error	Calc. Order
$FD1$	2.116	$6.27281 * 10^{-5}$	—
$FD1$	8.281	$1.61182 * 10^{-5}$	3.89176
$FD1$	32.761	$4.08124 * 10^{-6}$	3.94933
$FD2$	2.116	$2.58897 * 10^{-7}$	—
$FD2$	8.281	$4.55414 * 10^{-8}$	5.68487
$FD2$	32.761	$2.83388 * 10^{-9}$	16.07035
$PdeTbx$	2.129	$1.48329 * 10^{-4}$	—
$PdeTbx$	8.385	$3.76648 * 10^{-5}$	3.93812
$PdeTbx$	33.281	$9.48946 * 10^{-6}$	3.96912

Table 2: Error comparison on disk. Quadratic order for $FD1$, non-optimal starting grid for $FD2$.

[1] eqn = r D[u[r,t],t,1] == κD[rD[u[r,t],r],r];
bc = u[1,t] == 0; ic = u[r,0] == 1-r^2;
dsol=DSolve[{eqn,bc,ic},u[r,t],{r,t}]



$FD1$ 2.116 Pt

$FD1$ 8.281 Pt

$FD1$ 32.761 Pt

$FD2$ 2.116 Pt

$FD2$ 8.281 Pt

$FD2$ 32.761 Pt

$PdeTbx$ 2.129 Pt

$PdeTbx$ 8.385 Pt

$PdeTbx$ 33.281 Pt

Figure 7: Absolute errors $E_{i,j}(2)$ for different grid sizes on $\mathfrak{E}$.

Summary

The results of this paper show that combining the method of lines with conformal mappings is principally apt for numerical and/or symbolical treatment of PDEs. Moreover, the analytic solutions to problems on unit square and unit disk presented in this article are convenient for checking your own simulation code.

The developed simulation code in Mathematica can be regarded as a starting point from which one is able to march out and adapt it to his own needs: In keeping things modular, a diversity of higher-order derivative approximations can externally be implemented and loaded according to the desired error requirements (see the blue/bf coded section in the box).

On the other hand, a very broad range of PDE-classes are accessible by this approach (for time-dependent problems the bf/italic coded section is to be modified). While even tensor-valued state variables can be considered, the presented method is definitely restricted to two-dimensional geometries. This is caused by the fact that conformal transformations arise in complex analysis mapping $C \to C$.

So if we are able to construct a highly accurate conformal map from square to a more general domain [5] (aside the unit disk but potentially for convenience with the unit disk interconnected) and same holds for the numerical values of its derivatives, we would be in a good position to treat PDEs on such domains as well. This will be shown in a subsequent contribution.

In addition, another level of abstraction will be introduced by regarding physical laws in invariant PDE-formulation. The domains under consideration will then appear as two-dimensional (flat) Riemannian manifolds where the calculated derivatives of the conformal map are hence to be interpreted as metric quantities correcting the effect of using curvilinear co-ordinates: Fundamental tensor and Christoffel symbols will be used to realize the PDE in this 'conformal co-ordinates'.

In the presented method the conformal map played a double game: Grasping it as a point map we constructed the transformation from square to disk. On the other hand, to get a glimpse at the metric conditions, the map acted as transformation of co-ordinates supporting simulation - the disk has been parameterized by the square.

References

[1] Fornberg B. *Calculation of Weights in Finite Difference Formulas*. SIAM Rev., Vol. 40, No. 3, pp. 685-691, 1998, ISSN (print): 0036-1445.

[2] Hassenpflug WC. *Elliptic Integrals and the Schwarz-Christoffel Transformation*. Computers & Mathematics with Applications, Vol. 33. 15-114. 10.1016/S0898-1221(97)00091-6.

[3] Herod JV. *Asynchronous Studies in Undergraduate Partial Differential Equations. Lecture 33: The Heat Equation on a Disk*. https:/people.math.gatech.edu/∼herod/conted/M33.pdf

[4] Holzinger M. *Konforme Abbildungen zur Simulation von Modellen mit verteilten Parametern*. Dissertation/PhD (in German), Wien, 2020. http://katalog.ub.tuwien.ac.at/AC15652638

[5] Kythe PK. *Computational Conformal Mapping*. Birkhäuser, Boston 1998, ISBN 0-8176-3996-9.

[6] Schiesser WE. *The Numerical Method Of Lines. Integration of PDEs*. Academic Press Inc., San Diego, 1991, ISBN 0-12-624130-9.

[7] Trefethen LN, Driscoll TA. *Schwarz-Christoffel mapping in the Computer Era*. Proceedings International Congress of Mathematicians, Vol. III (Berlin, 1998), vol. 1998, pp. 533-542, ISSN 1431-0643.