

A few remarks on what I have done so far

Overview of Matlab Code — 17.05

Heat Equation, ie. $\frac{\partial u}{\partial t} - \Delta(u) = f$ with Dirichlet boundary conditions

- **parareal implementation**
- **sequential FD** implementation: "heat_eqn_seq_FD.m" solves the following approximation sequentially:

$$(I + \Delta t L_h)u_{i,j+1} = \Delta t f_{i,j+1} + u_{i,j}$$

- **sequential FEM** implementation: "heat_eqn_seq_FEM.m" uses L_h and M_h , and implicit Euler, solves LS either directly or using multigrid, i.e.

$$(M_h + \Delta t L_h)u_{i,j+1} = \Delta t M_h f_{i,j+1} + M_h u_{i,j}$$

- **space-time FD**: simplified Ganda & Neumueller using implicit Euler, all in big matrix
- **space-time FEM**: according to Ganda & Neumueller (really or not actually?) using implicit Euler, all in big matrix

Wave Equation, ie. $\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = f$

- **sequential solver**
- fully parallelised matrix form using 2nd order stencil in both time and space, needs solution for end time: "wave_eqn_parallel_1D.m" because we have entries on the upper triangular matrix, ie. boundary conditions needed
- matrix form using backward difference stencil of type $[1, -2, 1]$ for time derivative, "wave_eqn_like_NM.m", unfortunately DOES NOT seem to work, loses energy over time, not sure why
- version where I also use $[1, -2, 1]$ with the first set of 1's (times scalar) on diagonal, on lower diagonal block I then have a tridiagonals from the space stencil, for both cases need not only $u_0(x)$ but also next time step, can be obtained through $u'_0(x)$ though (which corresponds to matrix form of the sequential solver?!)

Functions to do set ups

1D:

- 2nd order finite difference stencil to approximate $\Delta(u)$: "set_up_L_h_FD.m"
- 2nd FEM stencil to approximate $\Delta(u)$: "set_up_L_h_FEM.m"
- FEM mass matrix, computes $\int \phi_i \cdot \phi_j dx$: "set_up_M_h_FEM.m"
- **for Wave Equation**: matrices K_h that act as if time was another space direction with special factor $-c^2$ in front, here I have "set_up_K_h.m" which is the equivalent to a 5 point stencil in 2D, I also have "set_up_K_h_backDiff.m" which uses a backward Differences of the form $[1 -2 1]$ where the first entry corresponds to the space entries we are solving for
- backward Euler
- V,W cycle multigrid, uses "mg_set_up.m" and above functions: "V_cycle.m", "W_cycle.m"

Open Questions

- Compare to Neumueller paper, where are the differences? Why does this seem to work fine in this setting? Difference heat, wave equation?

May 7th, 2018

I have mostly been writing little matlab programs, dealing either with the **Poisson, Heat Wave equation**.

- small parareal implementation of simple ODE
- parareal implementation of heat equation
- for wave equation, normal sequential solver, one simply pretending time was another dimension, problem here, need boundary conditions at the end, otherwise it does not work, so not very useful?
- most promising seeming one, 3 point stencil in space, backward difference in time, problem: system loses energy, when using a higher order stencil (I've been using 1st and 2nd order) then the system loses energy and slowly converges to zero (wave amplitude goes down ...) when decreasing time step size it gets better, but always seems to happen
- higher order stencil helps a bit but then we need 3 initial conditions, which is a problem as well (for first order stencil it is fine to just use u_0 twice but not for second order)
- what works best so far is simply the matrix version of the sequential solver implementation, does not seem to have the problem of losing energy, so this issue must arise from how the points in time and space are put together

June 15th, 2018

Characteristics ...

- thinking about level set method where the level sets are the characteristics of the pde
- what exactly are characteristics?
- so far LINEAR ADVECTION equation $\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0$, I rewrote it such that we consider it along its characteristics v_t and vector orthogonal to that v_n , then consider simple finite difference scheme in this new basis, used the following scheme to compute it
- and in comparison used this upwind scheme, that is apparently the most suitable one for $a > 0$, if I use i instead of $i - 1$ I get a lot of diffusion:

$$\frac{u_{i,j+1} - u_{i,j}}{h_t} + a \frac{u_{i,j} - u_{i-1,j}}{h_x} = 0$$

- the shape of u_0 seems to be maintained pretty well but for the computation along the axis the solution seems to be growing over time, question of stepsize or still a bug in the code?
- in normal scheme check for CFL condition, ie. $a \frac{\Delta t}{\Delta x} \leq 1$

July 23rd, 2018

Constructing an AMG solver for monodomain model

I'm considering the following equation in 2D for now

$$\partial_t V - \nabla \cdot (D(x) \nabla V) = I_{ion}(V) + I_{ext} \quad (1)$$

where $I_{ion} = V \cdot (a - V) \cdot (V - 1)$, $0 < a, 1$ and assume for now that $D(x) = I$, hence we get

$$\partial_t V - \Delta(V) = I_{ion}(V) + I_{ext} \quad (2)$$

which corresponds to a heat equation with a non-linear right hand side.

General questions to deal with:

- what is a realistic I_{ext} ?

Reformulate so that we get its weak formulation (taken from slides, just result now)

Notation: $E^m = [t_m, t_{m+1}]x\Omega$

$$-\int_{E^m} V \partial_t U \, dt dx + \int_{\Omega} [VU]_{t_m}^{t_{m+1}} \, dx + \int_{E^m} \nabla V \cdot \nabla U \, dt dx = \int_{E^m} f U \, dt dx \quad (3)$$

Here f not yet dependent on solution. What changes then?

Use Ganda & Neumueller approach to do space-time, we need to set up the different stencils and matrices, put them together with tensorproducts. Note that we are assuming that our basis and test functions are seperable in time and space in the sense that $V_h^m(x, t) = \sum_l \sum_k v_{l,k}^m \psi_l(x) \phi_k(t)$ when we insert them in the weak formulation. Overall we then get

$$\begin{aligned} A &= K_t \otimes M_h + M_t \otimes K_h \\ B &= -N_t \otimes M_h \end{aligned}$$

Build them with the help of Pietros matlab code and assemble them in the following way:

$$\begin{bmatrix} A & & & \\ B & A & & \\ & \dots & \dots & \\ & & B & A \end{bmatrix} \cdot \begin{bmatrix} u_{t_1} \\ \dots \\ \dots \\ u_{t_n} \end{bmatrix} = \begin{bmatrix} f_{t_1} \\ \dots \\ \dots \\ f_{t_n} \end{bmatrix}$$

How do my different stencils look like? Local mass matrix:

$$M_h = \frac{h^2}{24} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad L_h = \frac{1}{h^2} \begin{bmatrix} 4 & -1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \dots ?!$$

How does AMG work now:

STEP I: do ξ_1 smoothing steps

STEP II: compute fine grid residual $r^h = f^h - A^h v^h$ and restrict to coarse grid $r^{2h} = I_h^{2h} r^h$

INTERPOLATION OPERATOR $I_h^2 h$

notion of strong dependence ...

$$(I_{2h}^h e)_i = \begin{cases} e_i & \text{if } i \in C \\ \sum w_{ij} e_j & \text{if } i \in F \end{cases}$$

Continuation Monodomain — 01.08.18

Implementation 1D MATLAB

- general approach to solving this: LHS becomes one big space-time matrix M_{st} , RHS: write U and f in terms of basis functions, can restate as $(M_t \otimes M_h) \cdot (f \otimes [1, 1])$ or something like that
- how do we deal with $f(u)$, ie. f depending on u ? Just consider as pointwise evaluations?! use $f(u_k)$ as vector and keep the rest, linearised right hand side?
- let $M_{th} = (M_t \otimes M_h)$, $\hat{f} = (f \otimes [1, 1])$ and describe system as $M_{st}u = M_{th}\hat{f}(u)$
- solve for u using Newton:

$$\begin{aligned} J(u) &= M_{th}\hat{f}(u) - M_{st}u \\ H(u) &= J'(u) = M_{th}diag(\hat{f}'(u)) - M_{st} \\ u_{k+1} &= u_k - (H(u_k))^{-1}J(u_k) \end{aligned}$$

- next step would then be to not invert directly but instead to use AMG or whatever I come up with to solve

$$\begin{aligned} H(u_k)u_{k+1} &= H(u_k)u_k - J(u_k) \\ p(u_k) &= H(u_k)u_k - J(u_k) \end{aligned}$$

$$\mathbf{H}(\mathbf{u}_k)\mathbf{u}_{k+1} = \mathbf{p}(\mathbf{u}_k)$$

- or instead solve for the error instead of having u_k in equation again:

$$\begin{aligned} H(u_k)(u_{k+1} - u_k) &= J(u_k) \\ e_k &= u_{k+1} - u_k \end{aligned}$$

$$\begin{aligned} \mathbf{H}(\mathbf{u}_k)\mathbf{e}_k &= \mathbf{J}(\mathbf{u}_k) \\ u_{k+1} &= u_k + e_k \end{aligned}$$

Hence the matrix I'm really interested in is $H(u_k)$

- Otherwise if I'm doing smoothing steps I have the following iteration, let $H_k = H(u_k)$, $J_k = J(u_k)$:

$$e_{k+1} = e_k + B^{-1}(J_k - H_k e_k)$$

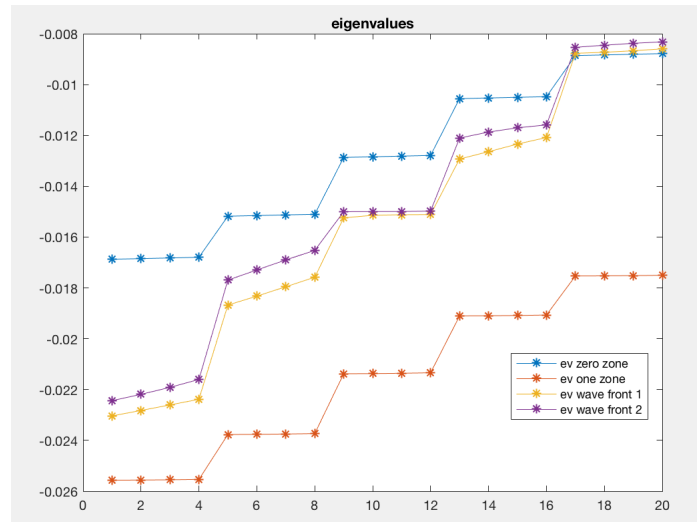
$$e_{k+1} = (I - B^{-1}H_k)e_k + B^{-1}J_k$$

Therefore we have the iteration matrix $(I - B^{-1}H_k)$, might want to look at that one as well? What to use as preconditioner ...

- How to handle initial conditions? $\hat{f}(u)(1 : 2Nx) = u_0 \cdot sth$ but what to do now for $\hat{f}'(u)(1 : 2Nx)$?!

Looking at Eigenvalues and Eigenvectors of H – 08.08.18

- check booklet for more theoretical analysis
- I have considered submatrices, in the sense that I chose certain points on the grid, for now I used the information that I already have about what is going on where ...
- GOAL: find differences between wavefront area and "flat" areas
- so I chose certain grid points (that give connected rectangles) and extracted the corresponding x_i – indices in H in time and space, I then get a list of indices in H , eg. $l = [15, 16, 17, 25, 26, 27, 35, 36, 37]$ for a 3×3 subdomain of a grid with 5 space components (get 10 because of $-$ and $+$ for all inner time points, would correspond to space indices $\{1, 2, 3\}$ and time indices $\{3, 4, 5\}$)
- then extract all values from H with $H(i, j)$, $i \in l, j \in l$, hence we get a 9×9 matrix, which means 9 eigenvalues and vectors
- – current understanding -09.08.- eigenvalues of one and zero zone always seem to come in pairs of 4 when there is 4 entries in time, and then there is 5 of those sets, which is the number of entries in space



Questions: For the one and zero zone we have always have sets which have the size of the number of points taken in time, ie. algebraic multiplicity of an eigenvalue corresponds to points taken in time, now look at what the corresponding eigenvectors look like:

- we can see that they correspond to one eigenfunction that is being shifted across the grid in time direction
- sometimes the sign is changed
- up until now the eigenvalues with the larger absolute value had eigenfunctions of lower frequency (as far as one can tell)
- if all eigenfunctions are some sort of sine or cosine combination, how would I find them explicitly?
- at first sight it seems like zero and one zone eigenfunctions are much more symmetric, whereas the wavefront ones seem to have non-symmetric sets
- since we only have so few gridpoints in there though, hard to estimate

AMG — 20/08/18

- set up of AMG, for first identity matrix used for Dirichlet boundary conditions, we have strong dependence on every other entry since $\max a_{ij} = 0$, in this case could make sense to simply define these nodes because don't depend on any other node, but then we also have no error that needs to be corrected because we already have exact solution here (probably only case where this can happen because if it didn't depend on anything and we didn't have solution how would we compute this value?), hence in this case it is appropriate that it is a fine grid node, but not appropriate that it depends on all other nodes
- What happens if we assign whole row to be zero, do we get a problem if S is singular? No, because S could be singular anyway
- but then i get a problem in assigning coarse and fine grid points, this point does not depend on anything, doesn't get assigned to fine grid points, should change that?! weird to implement but try it anyway to potentially get increase in speed
- potentially doesn't make sense, want these values in there for error correction, ie. might want them in coarse grid!?
- seem to have similar computational costs
- in case of grid of size 6 x 5 (6 pts in space, 5 in time) the support of the eigenfunction is over the entire space domain (actually seems cut off) and in time only one row of nonzero values, all other values zero

Looking at full submatrices of H, including double DG grid points — 30.09.18

- when we don't look at x_+, x_- but only one of them then the matrices are symmetric and there is no coupling between the timesteps ... but that is obviously not what we want, hence look at both entries
- problem: matrices not symmetric anymore: therefore no orthonormal basis of eigenvectors, we get complex eigenvalues and eigenvectors
- how do we plot them in a good way? one dimension too few (got space & time) as well as real and imaginary part

- and then how do we analyse them?

Observations so far:

- there do not seem to be any non-zero entries in lower part of grid, why is that? but according to matlab give full rank, so entries probably just very small
- did I organise them correctly? vectors that come out are in matrix V , where each column is an eigenfunction, first entry corresponds to lower left corner of the grid, want to write it into matrix, where does the entry a_{11} get plotted? $-i$ gets plotted at $x(1), y(1)$, therefore should work alright
- eigenvectors of constant areas seem very similar, I subtracted the different columns from each other and compared their norms, in a 40 by 40 system 32 of them have a "partner" in the other constant area, with a norm difference of $< 10^{-4}$ however some of the eigenfunctions are also incredibly similar to each other ...
- now compared 2 areas from constant zero zone, if I use a tolerance of 10^{-3} and check whether eigenfunction just multiplied by -1 then they all have a partner in the other set
- in comparison with a wave front set there are 0
- so now question would be: how early in iterative process can you see this? do constant 1 and constant 0 need to be treated differently somehow? how are your test grid patches set up? and what happens if they are half on wavefront half on constant area? probably not going to hit it straight in the middle? and then maybe most importantly, how do I decide which eigenvectors on wavefront are the ones I should choose? which ones are the most important ones in shaping the solution? also how do I deal with real and imaginary part?