Let's View Higher Dimensional PDE in Tensor Domain

t-CUR,M-CUR Based Video Completion

Author: Suvendu Kar

M.Tech. CDS(IISc Bangalore)

suvendukar@iisc.ac.in

Table of Contents

Department of Computational and Data Sciences

Higher Dimensional Generalization of PDE into Tensor

We are in the era of Deep-Tech.Base of fascinated NNs are matrix computation

Often we need face to solve numerical PDEs in applied mathematics,fluid dynamics, computational physics,space theory,etc.

More grid points while discretizing is fine but increase complexity.Thanks to Tensor,for reducing time/computation complexity.

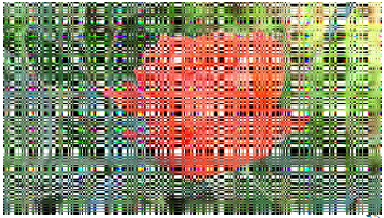
Are you interested to know more?

CDS
Department of Computational and Data Sciences

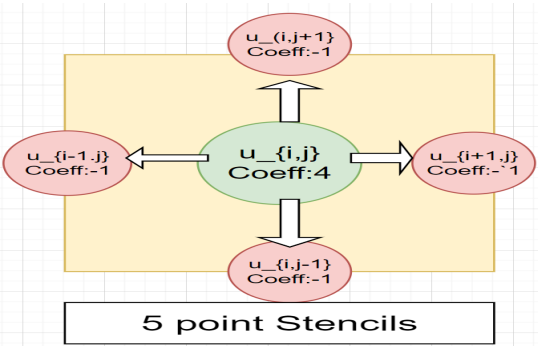
Video Recovery

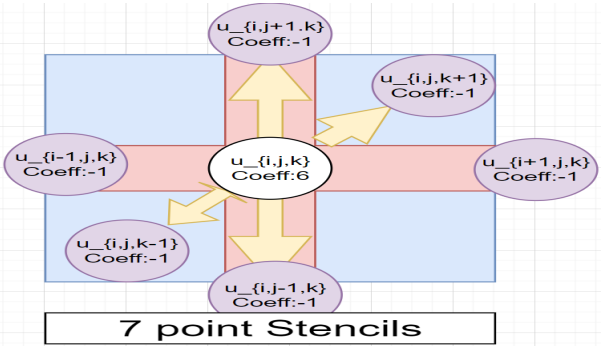
t-CUR,M-CUR

t-CUR,M-CUR based video recover from corrupted frames.



Mathematical Understanding of 5 point and 7 point stencils





CDS
Department of Computational and Data Sciences

Mathematical Understanding for PDE Solving

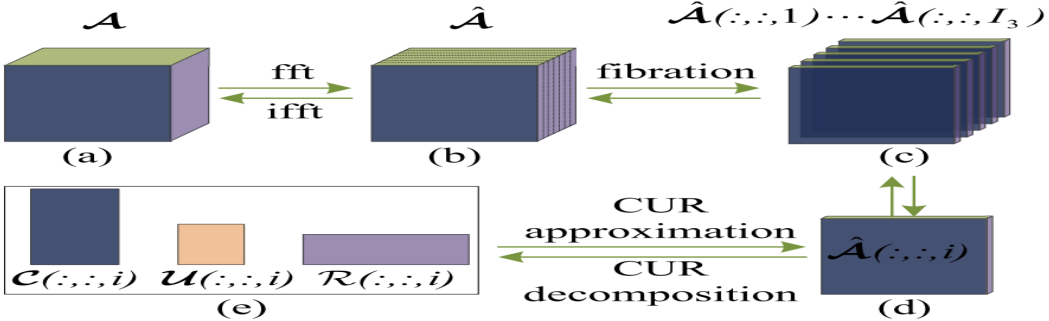
- Require Mathematical concepts that Used for PDE Solver
  - Reducing 2D Poisson Equation to Tensor based System of Equation
  - Reducing 3D Poisson Equation to Tensor based System of Equation
  - Einsetin product based Pseudoinverse Finding
  - Gauss Jacobi Method to Solve

Here is more details.

Go to Mathematical Formulation

CDS
Department of Computational and Data Sciences

Mathematical Understanding of t-CUR



Mathematical Understanding for Video Recovery

- Require Mathematical concepts that Used for Video Recovery
  - T-CUR Based Tensor Completion
  - M-CUR Based Tensor Completion

Here is more details.

Go to Mathematical Formulation

CDS
Department of Computational and Data Sciences

No other way of simplified PDE Solving??

1. Do you aware of PINN?
   Meshfree alternative of traditional approach.Automatic differentiation(autograd) is the power here.
2. Easy to implement and very advanced to find numerical solutions even with low number of training data points.

For PDE Solver

Direct Method

Direct method i.e. Pseudoinverse based, is good for low N, but time consuming for higher N.With low N, errors are near to 10^{-14}.

Iterative Methods

Iterative solver (Gauss Siedel) is very timeefficeint and error decay polynomially with the increment of N.

PINN Based Method

With even low data points (sometime even without knowing boundary conditions) achieves high accuracy.Both time and space efficeint.

Go to Experimental Observation

For Video Reconstruction

MCUR vs tCUR for Error

With same number of sampled row and column, error(frobenius norm of approximated result-actual result)in MCUR based approach is higher than tCUR based approach.

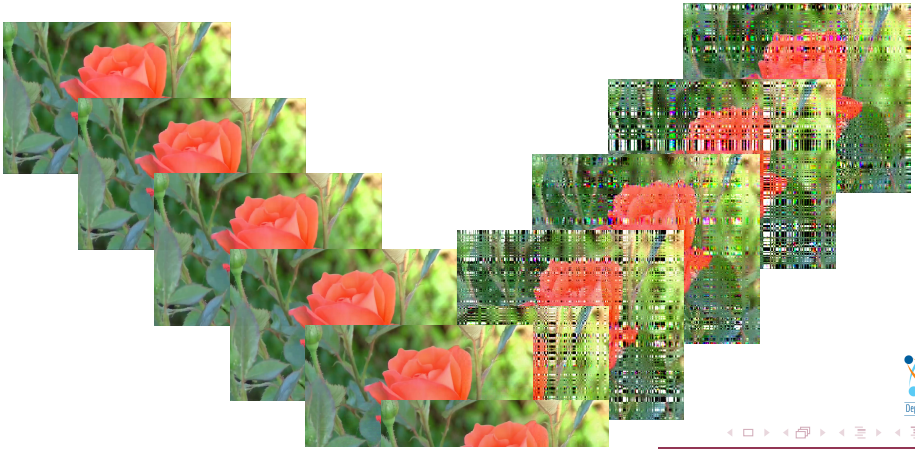
MCUR vs TCUR

Also MCUR based approach is relatively higher in time consuming.

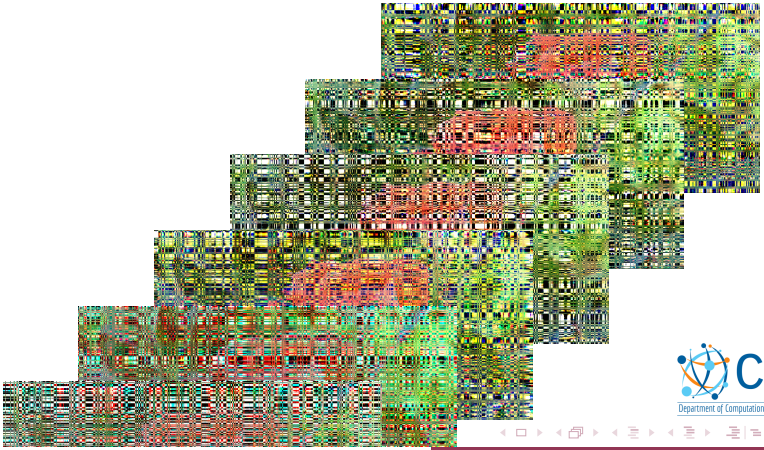
Go to Experimental Observation

CDS
Department of Computational and Data Sciences

For Video Reconstruction comparison with number of rows=columns=100, with t–CUR

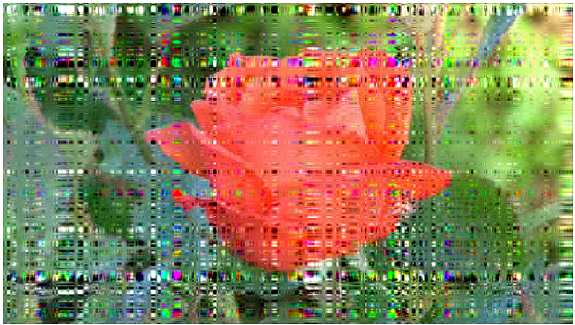


For Video Reconstruction with number of rows=columns=100, with m-CUR

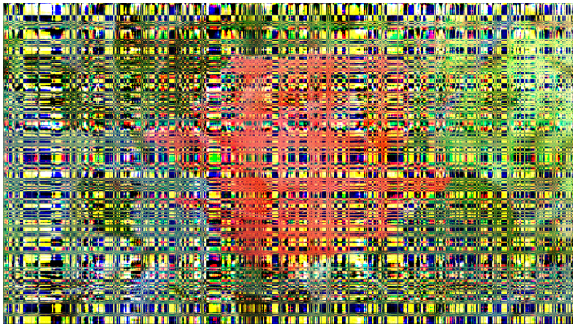


Actual Video

Reconstructed video with t-CUR and 700 rows,column



Reconstructed video with m-CUR and 700 rows,column



Conclusion!!

- Experiments on solving higher dimension PDE with both direct and iterative solver gave an idea how tensor can help us to overcome matrix-computation based complexity.Overall, for lower grid points direct t solver is good , for higher grid points iterative solver is good.

- Video recovery (4 way tensor completion)was a valid extension of 3-way tensor completion problem.t-cur did good job compare to m-cur.

CDS
Department of Computational and Data Sciences

Questions?

References I

1)M. Brazell, N. Li, C. Navasca, et al. Solving multilinear systems via tensor inversion. SIAM J. Matrix Anal,Appl. 2013;34(2):542-570.
2)Further results on the Drazin inverse of even-order tensors;Ratikanta Behera, Ashish Kumar Nandi,Jajati Keshari Sahoo,WILEY,DOI: 10.1002/nla.2317
3)DS 285 Class Note

CDS
Department of Computational and Data Sciences

Appendix - 2D Poisson Equation

Return to presentation 2D Poisson's equation on $[0,1] \times [0,1]$:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 2 * \pi^2 * sin(\pi * x) * sin(\pi * y) \text{ in } \Omega.$$
$$u = 0 \text{ on } \partial\Omega.$$

Let,$\mathbf{A} = tridiag(-1, 2, -1)$.Using 5 point stencils equivalent tensor formulation is:$\mathscr{A} *_2 \mathbf{X} = \mathbf{F}$, where $\mathscr{A} = reshape(\mathbf{A} \bigotimes \mathbf{I} + \mathbf{I} \bigotimes \mathbf{A}, [N, N, N, N])$,and $\mathbf{F} = reshape(vec(f_{ij}), [N, N])$.

CDS
Department of Computational and Data Sciences

Appendix - 3D Poisson Equation

Return to presentation 3D Poisson's equation on $[0,1] \times [0,1] \times [0,1]$:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} - \frac{\partial^2 u}{\partial z^2} = 3 * \pi^2 * sin(\pi * x) * sin(\pi * y) * sin(\pi * z) \text{ in } \Omega.$$

$$u = 0 \text{ on } \partial\Omega.$$

Let,$\mathbf{A} = tridiag(-1, 2, -1)$.Using 7 point stencils equivalent tensor formulation is:$\mathscr{A} *_3 \mathbf{X} = \mathbf{F}$, where
$\mathscr{A} = reshape(\mathbf{A} \bigotimes \mathbf{I} \bigotimes \mathbf{I} + \mathbf{I} \bigotimes \mathbf{A} \bigotimes \mathbf{I} + \mathbf{I} \bigotimes \mathbf{I} \bigotimes \mathbf{A}, [N, N, N, N, N, N])$,and
$\mathbf{F} = reshape(vec(f_{ijk}), [N, N, N])$.

Appendix - Einstein Product Based Pseudoinverse

Return to presentation

Algorithm Pseudo inverse of 4th order tensor

function PINV4(A)
  $szA \leftarrow$ size of A ▷ Finding the size of A
  $B \leftarrow$ reshape$(A, [szA(1) * szA(2), szA(3) * szA(4)])$ ▷ Reducing A to the equivalent matrix B
  $[Um, Sm, Vm] \leftarrow$ svd(B) ▷ Getting the SVD of B
  for $i = 1$ to min(size$(Sm, 1)$, size$(Sm, 2)$) **do**
    if $Sm(i, i) \neq 0$ **then**
      $Sm(i, i) \leftarrow 1/Sm(i, i)$
    else
      $Sm(i, i) \leftarrow Sm(i, i)$
    end if
  end for
  $Um \leftarrow Um'$ ▷ Computing Um' to find pseudoinverse of B
  $U_{\text{inv}} \leftarrow$ reshape$(Um, [szA(3), szA(4), szA(1), szA(2)])$ ▷ Using *reshape* operation to get U_{inv}
  $V_{\text{inv}} \leftarrow$ reshape$(Vm, [szA(3), szA(4), szA(1), szA(2)])$ ▷ Using *reshape* operation to get V_{inv}
  $S_{\text{inv}} \leftarrow$ reshape$(Sm, [szA(1), szA(2), szA(3), szA(4)])$ ▷ Using *reshape* operation to get S_{inv}
  return $U_{\text{inv}}, S_{\text{inv}}, V_{\text{inv}}$
end function

Appendix - Gauss Jacobi Solver

Return to presentation

Higher-Order Jacobi Method

Given $A \in \mathbb{R}^{N \times N \times N \times N}, B \in \mathbb{R}^{N \times N}, MAX$
Initial guess $X^0 \in \mathbb{R}^{N \times N}$
for $k = 1$ to MAX
  for $i = 1$ to N
    for $j = 1$ to N
      $X_{ij}^{(k+1)} = (B_{ij} - \sum_{i'j' \neq ij}(A_{iji'j'} X_{i'j'}^{(k)}))/A_{ijij}$
    end
  end
end

Return to presentation

t-CUR Based tensor completion

Algorithm :Tensor-TCUR Approximation From Partially Observed Entries (T-CUR)

Require: Observation data $P_\Omega(\mathscr{A}) \in R^{I_1 \times I_2 \times I_3}$, sampled columns/rows number di/li.Random Sampling has been used.

1. $\hat{\mathscr{A}} = FFT\left(P_\Omega(\mathscr{A}), [], 3\right)$;
2. for $i \leftarrow 1, \ldots, I_3$ do
3. $\left[\hat{\mathscr{C}}^{(i)}, \hat{\mathscr{U}}^{(i)}, \hat{\mathscr{R}}^{(i)}\right] = \text{MATRIX-CUR}\left(P_\Omega\left(\hat{\mathscr{A}}^{(i)}\right), d_i, I_i\right)$

4.end for

5.$\mathscr{C} = \text{IFFT}(\hat{\mathscr{C}}, [], 3); \mathscr{U} = \text{IFFT}(\hat{\mathscr{U}}, [], 3); \mathscr{R} = \text{IFFT}\ (\hat{\mathscr{R}}, [], 3)$

6.Return $\mathscr{C}, \mathscr{U}, \mathscr{R}$ such that a tensor cur approximation of $\mathscr{A} : \mathscr{Y} = \mathscr{C} * \mathscr{U} * \mathscr{R} \in R^{I_1 \times I_2 \times I_3}$

CDS
Department of Computational and Data Sciences

Return to presentation

M-CUR Based tensor completion

Algorithm : Tensor-MCUR Approximation From Partially Observed Entries. Require: Observation data $P_\Omega(\mathscr{A}) \in R^{I_1 \times I_2 \times I_3}$, vector of sample columns/rows number di/Ii

1. $\hat{\mathscr{A}} = mat\,(P_\Omega(\mathscr{A}), [], 3)$;
2. for $i \leftarrow 1, \ldots, I_3$ do
3. $\left[\hat{\mathscr{C}}^{(i)}, \hat{\mathscr{U}}^{(i)}, \hat{\mathscr{R}}^{(i)}\right] = \text{MATRIX-CUR}\left(P_\Omega\left(\hat{\mathscr{A}}^{(i)}\right), d_i, I_i\right)$

4.end for

5.$\mathscr{C} = \text{mat}^{-1}(\hat{\mathscr{C}}); \mathscr{U} = \text{mat}^{-1}(\hat{\mathscr{U}}); \mathscr{R} = \text{mat}^{-1}\,(\,\hat{\mathscr{R}})$

6.Return $\mathscr{C}, \mathscr{U}, \mathscr{R}$ tensor cur approximation of $\mathscr{A} : \mathscr{Y} = \mathscr{C} *_M \mathscr{U} *_M \mathscr{R} \in R^{I_1 \times I_2 \times I_3}$

CDS
Department of Computational and Data Sciences

Appendix - Observation on Iterative and direct 2D,3D PDE solver

Return to presentation E1=Fronius norm($\mathscr{A} *_k \mathbf{X}_{Predicted} - \mathbf{F}$); E2=Frobenius norm($\mathbf{X}_{Prediction} - \mathbf{X}_{True}$)

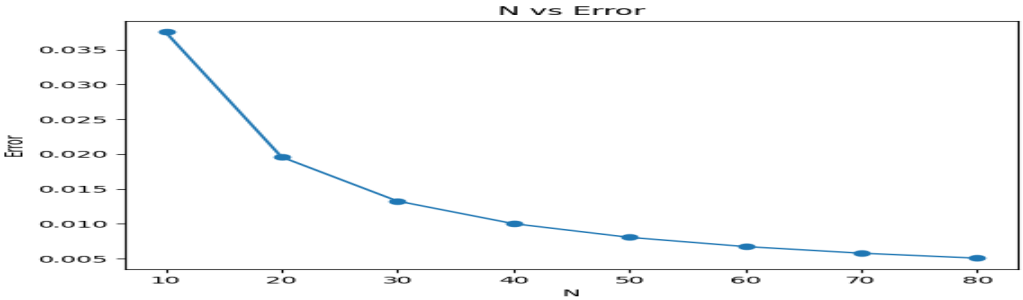


Figure: N vs Error E2 for Pseudoinverse based 2D Poisson Equation Solver

Appendix - Observation on Iterative and direct 2D,3D PDE solver

Return to presentation

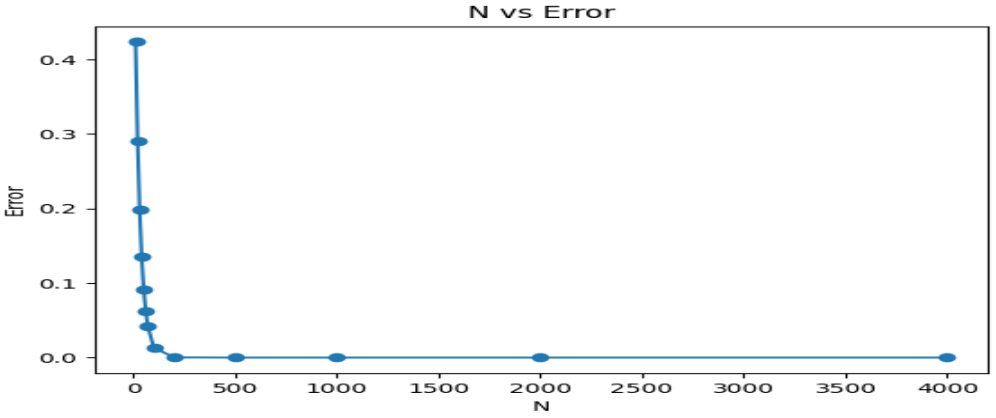


Figure: Iteration count vs Error E1 with Jacobi Method for 2D Poisson Problem

Appendix - Observation on Iterative and direct 2D,3D PDE solver

Return to presentation

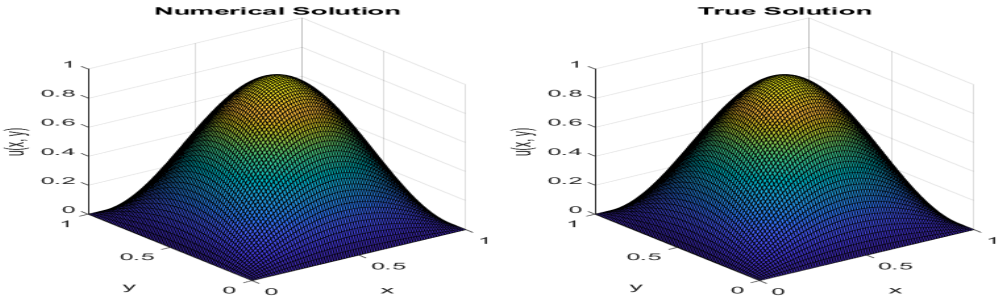
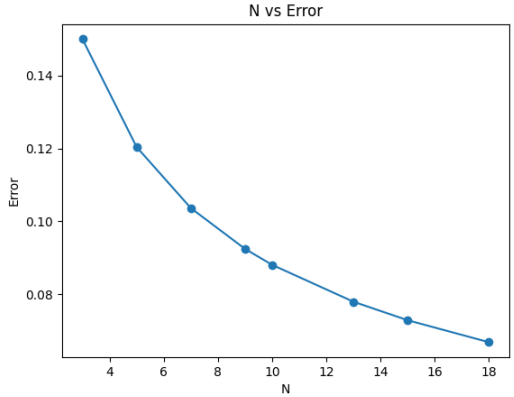


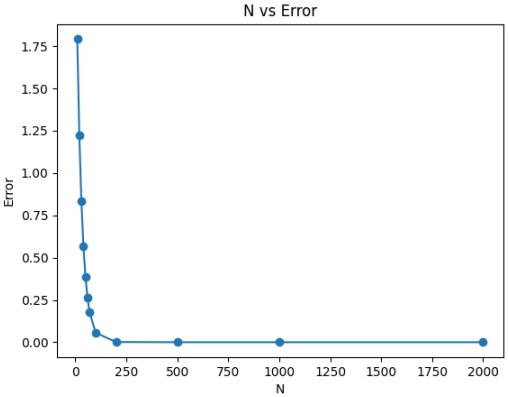
Figure: True solution vs Predicted Solution for N=80, for Poisson 2D Problem.

CDS
Department of Computational and Data Sciences

Appendix - Observation on Iterative and direct 2D,3D PDE solver

Return to presentation





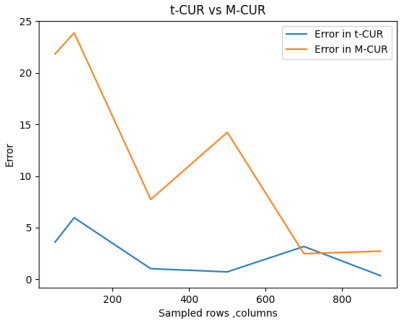
(a) N vs E2 Error with Pseudoinverse method for 3D Poisson Equation

(b) Iteration vs E1 Error with Jacobi method for 3D Poisson Equation

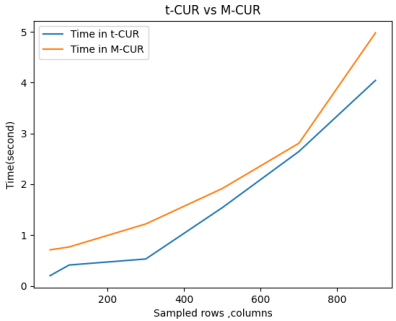
CDS
Department of Computational and Data Sciences

Appendix - Video reconstruction using tcur and mcur

Return to presentation



(a) a



(b) b

Figure: a):Error in t-cur vs m-cur (b):time taken in t-cur vs m-cur

CDS
Department of Computational and Data Sciences