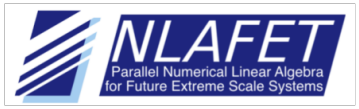
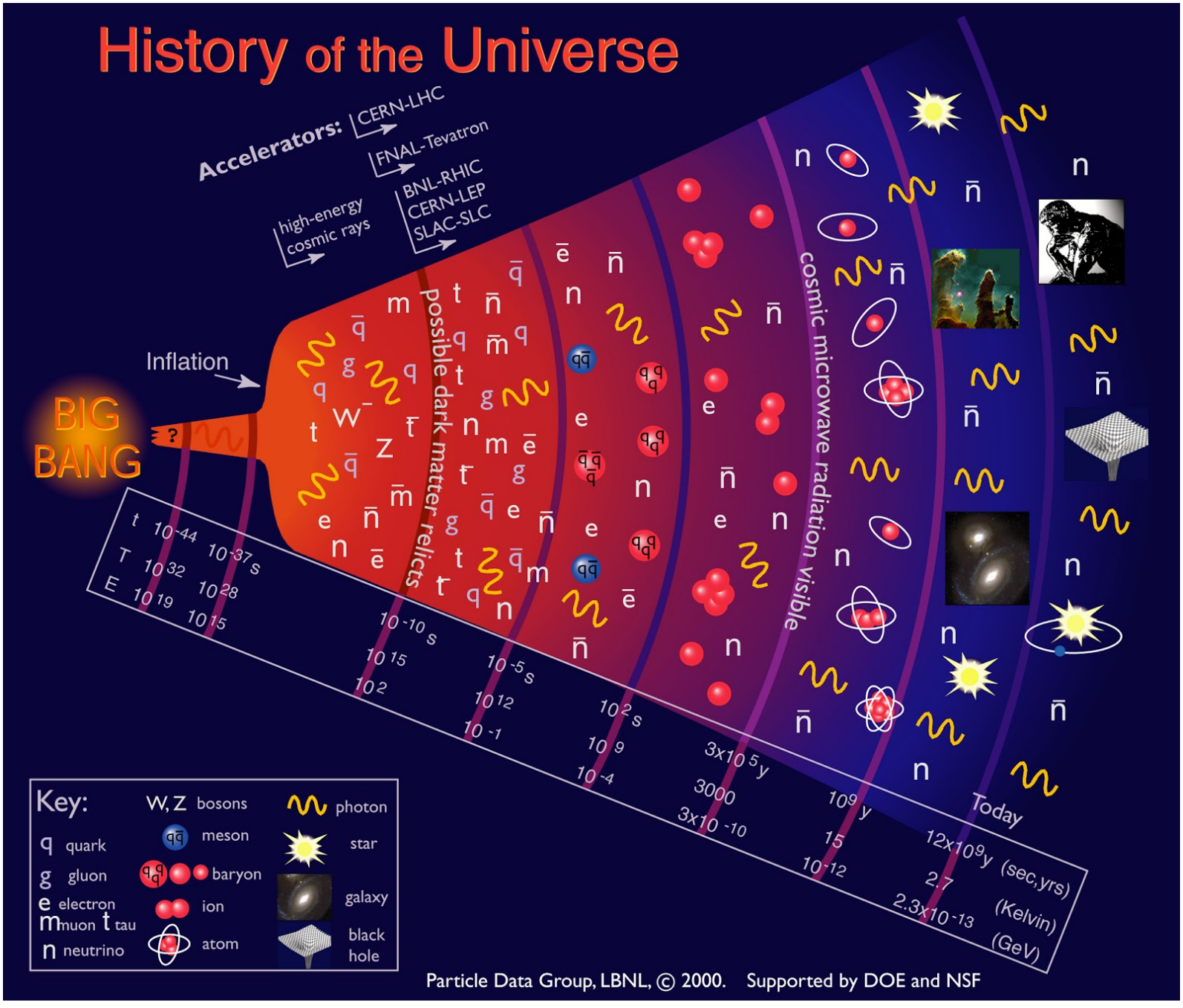
Preconditioned linear solvers in CMB data analysis

Jan Papež
Inria Paris, Alpines

Sparse Days 2018 @ Cerfacs
September 28, 2018

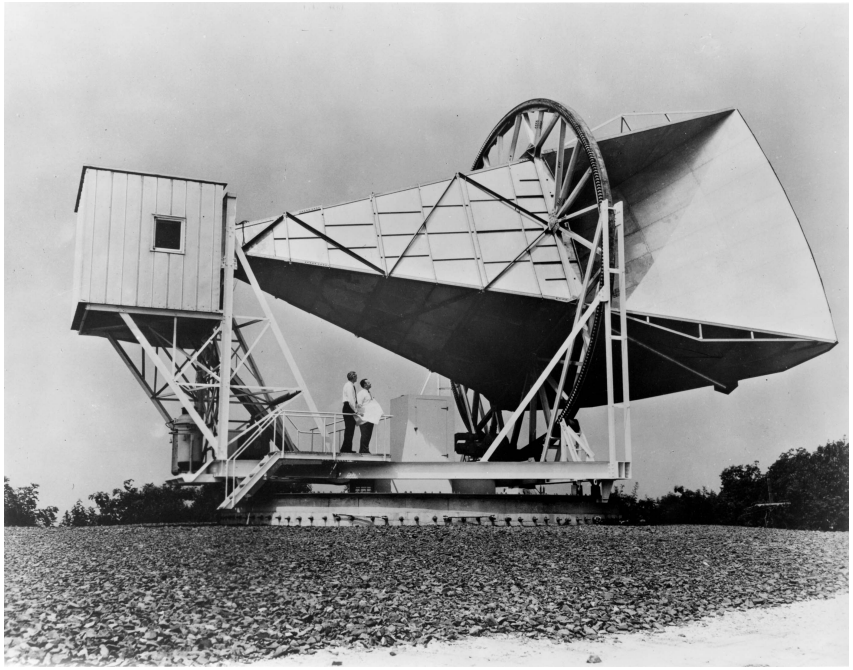



History of the Universe
Accelerators:
CERN-LHC
FNAL-Tevatron
BNL-RHIC
CERN-LEP
SLAC-SLC
high-energy cosmic rays
Inflation
BIG BANG
possible dark matter relicts
cosmic microwave radiation visible
t 10-44 10-37 s
T 1032 1028
E 1019 1015
10-10 s
10 15
10 2
10-5 s
10 12
10 -1
102 s
10 9
10 -4
3x10 5 y
3000
3x10 -10
109 y
15
10-12
Today
12x109 y (sec,yrs)
2.7 (Kelvin)
2.3x10 -13 (GeV)
Key:
q quark
g gluon
e electron
m muon t tau
n neutrino
W, Z bosons
meson
baryon
ion
atom
photon
star
galaxy
black hole
Particle Data Group, LBNL, © 2000. Supported by DOE and NSF

Cosmic microwave background (CMB) - brief history

1948 predicted by Ralph Alpher and Robert Herman

1964 accidentally discovered (measured) by Arno Penzias and Robert Wilson

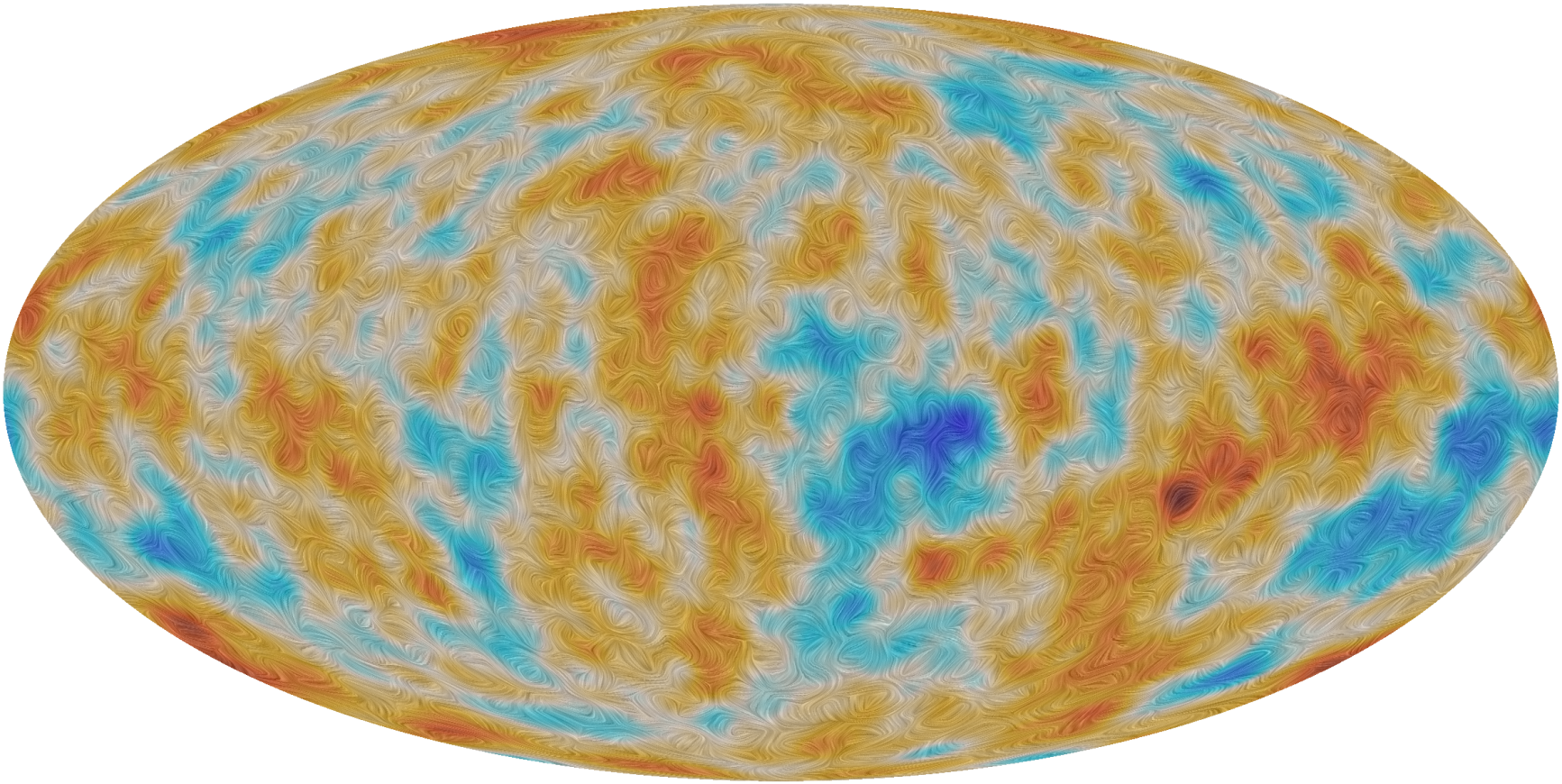


1978 Penzias and Wilson received the Nobel Prize in Physics

1992 results of NASA COBE experiment published

2006 Nobel Prize in Physics for COBE team

CMB - measured data



A visualization of the polarization of CMB as detected by ESAs Planck satellite over the entire sky.

Outline

Introduction – Cosmic microwave background (CMB)

Map-making in CMB

Linear solvers for map-making

Solving a sequence of (SPD) systems

Conclusions

this is a joint work with T. Cimic, L. Grigori, R. Stompor, and O. Tissot

Map-making in CMB

$$\underbrace{d}_{\text{observed data}} = \underbrace{P}_{\text{pointing matrix}} \cdot \underbrace{m}_{\text{unknown map}} + \underbrace{n}_{\text{noise}}$$

Maximum likelihood estimate:

$$(P^T N^{-1} P) m_{mm} = P^T N^{-1} d, \qquad N \ldots \text{covariance matrix of } n.$$

- noise is large (w.r.t. data), we assume that its covariance is known and it can be sufficiently tightly approximated by a *circulant* matrix
- the system matrix $P^T N^{-1} P$ cannot be, in general, formed but we can perform matrix-vector product with it:

$$(P^T N^{-1} P) v = P^T \cdot \left(IFFT \left(\Lambda_{N^{-1}} \cdot \left(FFT(P \cdot v) \right) \right) \right)$$

Maps and time data

System matrix: $P^T N^{-1} P$

Pointing matrix

P: map $\mapsto$ time data

at each time we observe one "pixel" $\Rightarrow$ P is very sparse

Size of the data

anticipated in future: maps $\approx O(10^6 - 10^8)$, time data up to $O(10^{15})$

in our experiments: map $= 3 \times 65\,536$, time data $= 2 \times 67\,108\,864$

$\Longrightarrow$ operations with a signal are negligible w.r.t. applying N^{-1}

Map-making in CMB II

Solving a linear algebraic system is (typically) only the first step in more complex procedure that leads to estimation of few cosmology parameters

$$\Longrightarrow \text{ we require an accurate and fast solution}$$

Interest in

- solution of a given (single) system
- solving a sequence (set) of systems (everyday measurements, divide-and-conquer-like approaches, parameter estimation $\rightarrow$ Monte Carlo simulations, . . .)

$$? \quad \boxed{\begin{array}{c}\text{simple preconditioner} + \text{more iterations} \\ \times \\ \text{more complex (and costly) preconditioner} + \text{less iterations}\end{array}} \quad ?$$

Solving a single system in CMB

State-of-the-art solver is PCG with block-diagonal preconditioner ($O(10)$ – $O(100)$ iterations). Other choices:

- two-level preconditioner ([Grigori, Stompor, Szydlarski 2012, 2014])
- pseudo-inverse preconditioner (idea $P^\dagger N (P^\dagger)^T \approx \left(P^T N^{-1} P\right)^{-1}$, inspired by [Elman 1999])
- multilevel preconditioner ([Seljebotn et al, 2017])
- messenger field with cooling procedure ([Elsner & Wandelt 2013]: *Efficient Wiener filtering without preconditioning*)
- enlarged PCG

Messenger field method (MF)

claims in the original paper

- approximation converges to the solution, $m^{(i)} \to m_{mm}$
- method is unconditionally stable and the convergence is exponential

while

If we write the original problem as $Ax = b$, MF is based on the splitting

$$A = C - D = C(I - C^{-1}D)$$

with an invertible C such that C^{-1} is easily computable. This is then used to define iterative (fixed-point) scheme

$$x^{(i+1)} = C^{-1}Dx^{(i)} + C^{-1}b.$$

Cooling technique (for messenger field)

Replace $Ax = b$ by

$$A_\lambda x_\lambda = b_\lambda, \tag{1}$$

such that

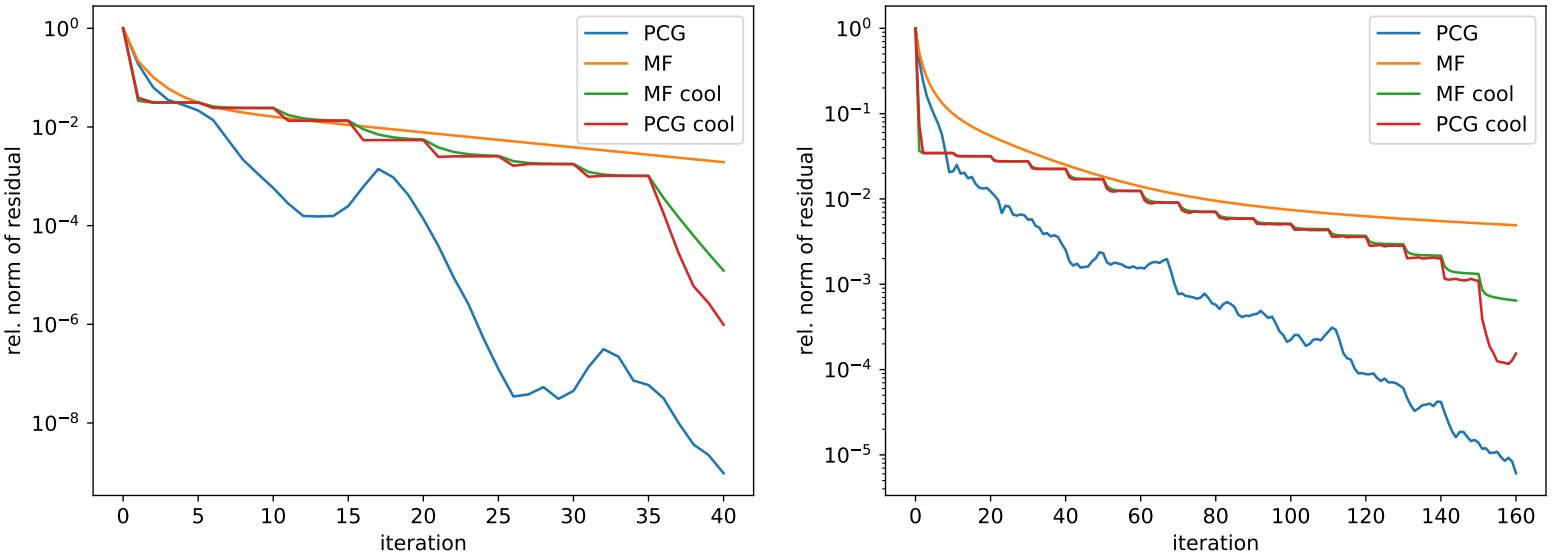
- $A_1 = A$, $b_1 = b$;
- $A_\lambda = C_\lambda - D_\lambda = C_\lambda \left(I - C_\lambda^{-1} D_\lambda\right)$;
- $C_\lambda^{-1} A_\lambda x_\lambda = C_\lambda^{-1} b_\lambda$ is easy to solve for $\lambda \gg 1$;
- $x_\lambda \to x$ as $\lambda \to 1$.

Then,

1. pick $\lambda \gg 1$, set $\widetilde{x} = 0$
2. solve (1) by messenger field iterating on $\widetilde{x}$
3. IF $\lambda = 1$, STOP; ELSE decrease λ and go to 2.

Numerical results – (cooled) MF vs. PCG

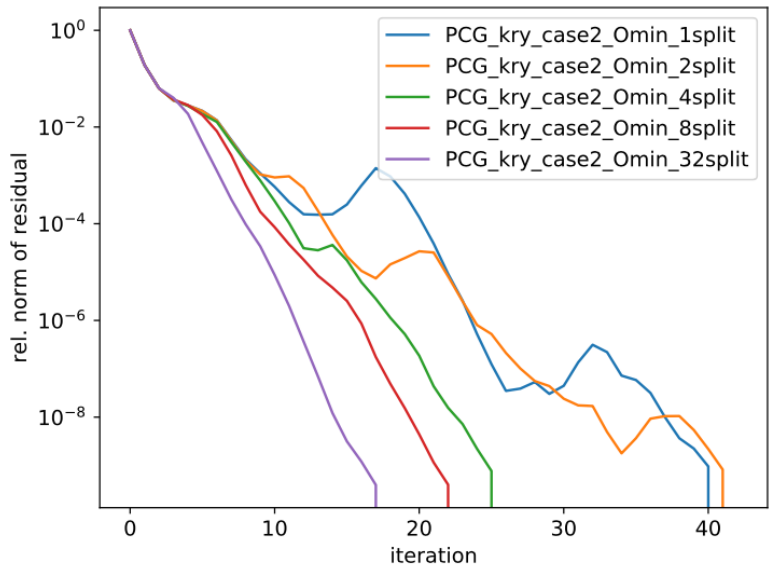
However, in our experiments,

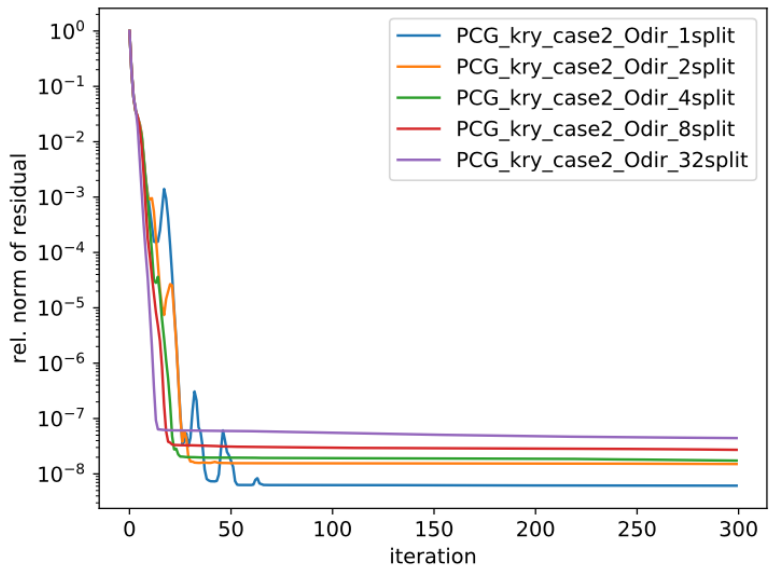


relative residual in two test cases

Enlarged PCG in map-making – `Odir` × `Omin`

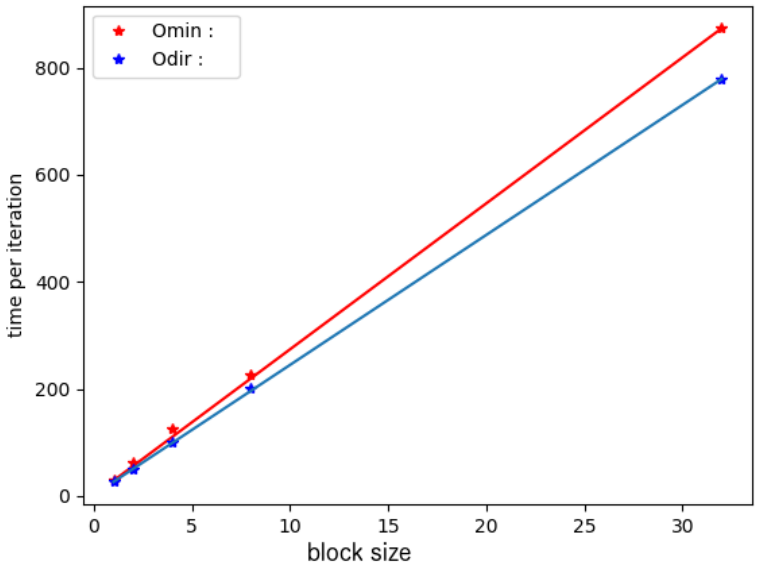
- in some experiments, `Omin` breaks down; a matrix that is assumed to be SPD is numerically singular → *this is understood behaviour and it can be avoided by an additional QR decomposition*
- maximal attainable accuracy: `Odir` saturates at level 10^{-8} – 10^{-6}
  → *this is not yet described in literature and we search for an explanation*





Enlarged PCG in map-making – block FFT

- FFT is the most costly part of each iteration
- currently, [cost of FFT for block] $\approx$ [cost of vector FFT $\times$ blocksize]
- straightforward first-level parallelization in size of the block
- ?effective parallel block-FFT ?

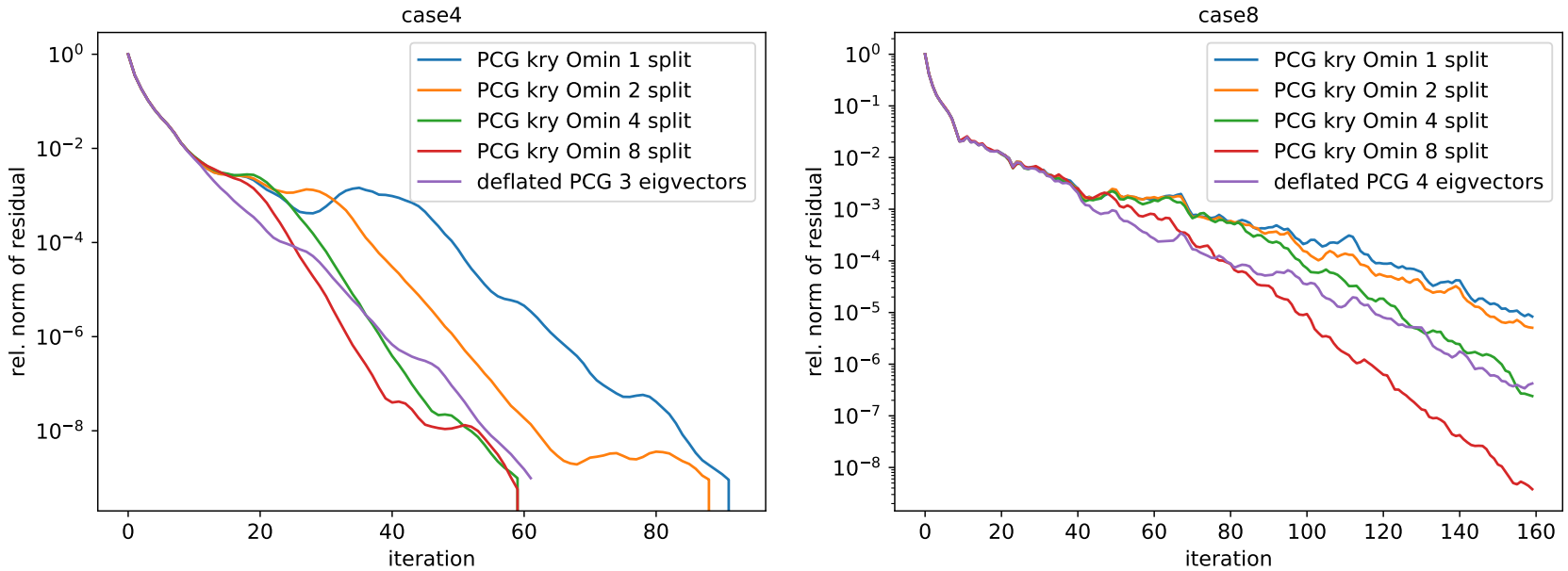


blocksize	1	2	4	8	32
# iter	65	50	39	34	25

Time of one iteration in current implementation and number of iterations to reach convergence

Enlarged PCG x deflated PCG in map-making

Comparison with deflated PCG (= two-level preconditioner), which requires computation of the "slowest" eigenvectors



relative residual in two test cases (`Omin` variant)

After initial iterations, it seems that deflated PCG with k eigenvectors converges similarly to enlarged PCG with size of the block $= k$.

Solving a sequence of (SPD) systems

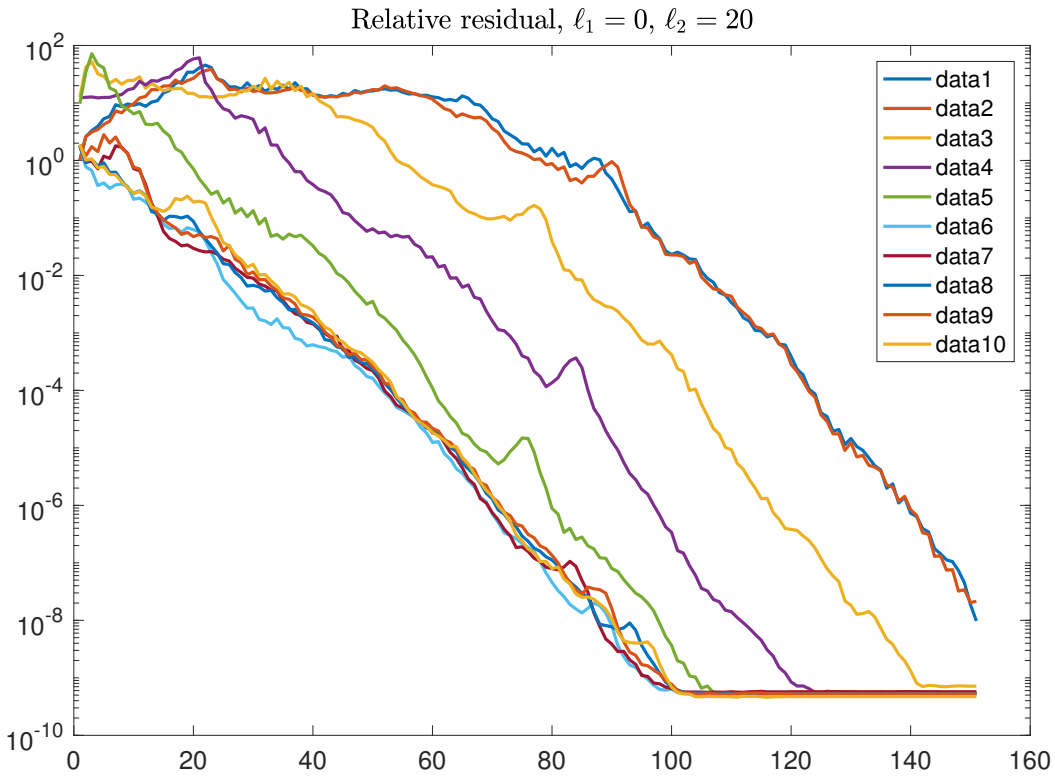
possible applications: CMB, Newton method, time-dependent PDEs, modelling fatigue and fractures, . . . cf. the block Krylov methods!

Idea:

- using GMRES/PCG with deflation and
- "smoothing" preconditioner (ILU, block Jacobi, . . .) and
- recycling the information from the Krylov subspace build in GMRES/PCG iterations for updating the approximation of the "slowest" eigenvectors

[Saad et al. 2000], [Morgan 2002], [Parks et al. 2006], [Jolivet & Tournier 2016]; in CMB (without recycling): [Puglisi et al. 2018]

Solving a sequence of (SPD) systems – illustration



a result from [Saad et al. 2000] for the sequence with constant system matrix

Ideally, we recover the slowest eigenvectors within solution of the systems.

Solving a sequence of (SPD) systems II

Let the systems to solve are denoted as

$$A^{(s)}x^{(s)} = b^{(s)}, \qquad s = 1, \ldots, S$$

Illustrative scheme:

given the basis $W^{(0)}$ of the deflation space (possibly empty),

1. a priori adapt the deflation vectors, e.g. using rank-revealing QR
2. run GMRES/PCG with a given preconditioner and deflation subspace given by $W^{(s)}$, save some of the Arnoldi/Lanczos vectors in V
3. a posteriori adapt $W^{(s)}$, e.g. using the (harmonic) Ritz approximation with $[W^{(s)}, V]$

The work (application of the ideas in CMB map-making) is currently in progress, together with Olivier Tissot.

Conclusions

- CMB data analysis is a challenging application
- it has some specifics, which can make even simple ideas applicable (and effective)
- application to real data requires powerful HPC resources (Cori machine, #10 in top500) and high HPC proficiency
- ? parallel FFTs for a block of vectors ? $\rightarrow$ enlarged PCG, deflation

Conclusions

- CMB data analysis is a challenging application
- it has some specifics, which can make even simple ideas applicable (and effective)
- application to real data requires powerful HPC resources (Cori machine, #10 in top500) and high HPC proficiency
- ? parallel FFTs for a block of vectors ? $\rightarrow$ enlarged PCG, deflation

Thank you for your attention