

JULY 6-10, 2015

ICC-JEJU

JEJU ISLAND, KOREA

DD XXIII

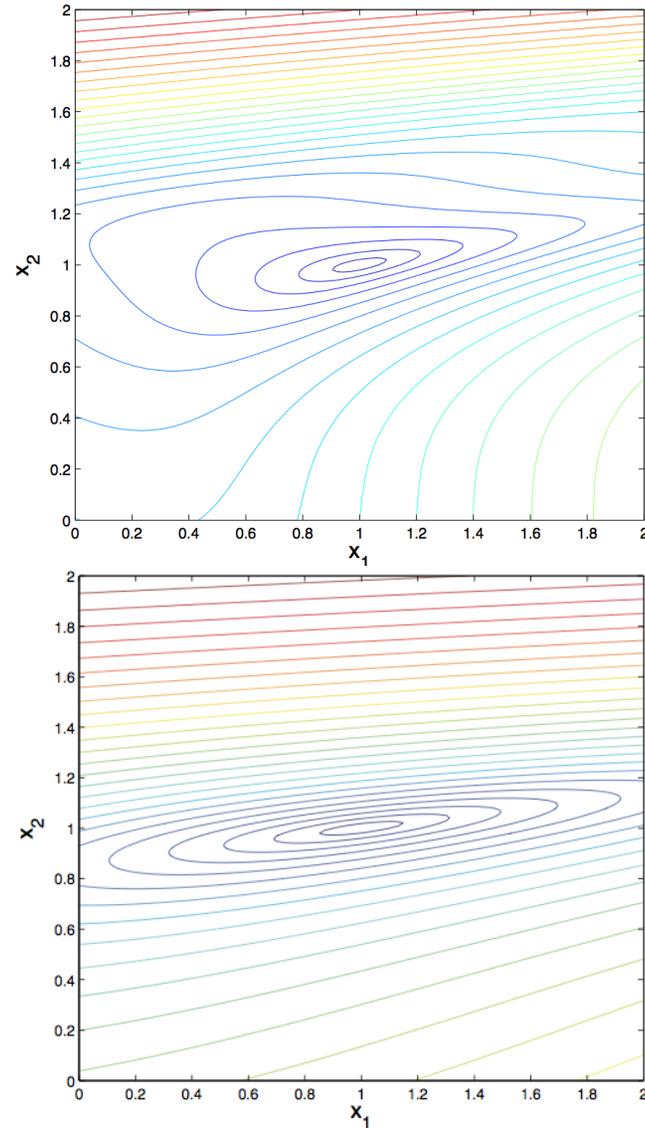
INTERNATIONAL CONFERENCE ON  
DOMAIN DECOMPOSITION METHODS

# Nonlinear Schwarz Preconditioning

**Lulu Liu & David Keyes**  
**Extreme Computing Research Center**  
**KAUST**

# The problem

- A nonlinear system  $F(u) = 0$  may be “stiff,” in the sense that the isocontours of the merit function, e.g.,  $f(u) = \|F(u)\|^2$ , are far from hyperellipsoidal, giving a small local convergence domain
- This may be combined with linear ill-conditioning, in the sense that the hyperellipsoids are locally badly stretched



# Typical causes of nonlinear stiffness

[Cai, K, Young, 2000] : **shocks, reaction zones, boundary layers, interior layers**

$$(A\rho u_x)_x = 0, \quad 0 < x < 2$$

$$A = A(x) = 0.4 + 0.6(x - 1)^2,$$

$$\rho = \rho(v) = (c^2)^{1/(\gamma-1)} = \left(1 + \frac{\gamma-1}{2}(1-v^2)\right)^{1/(\gamma-1)}$$

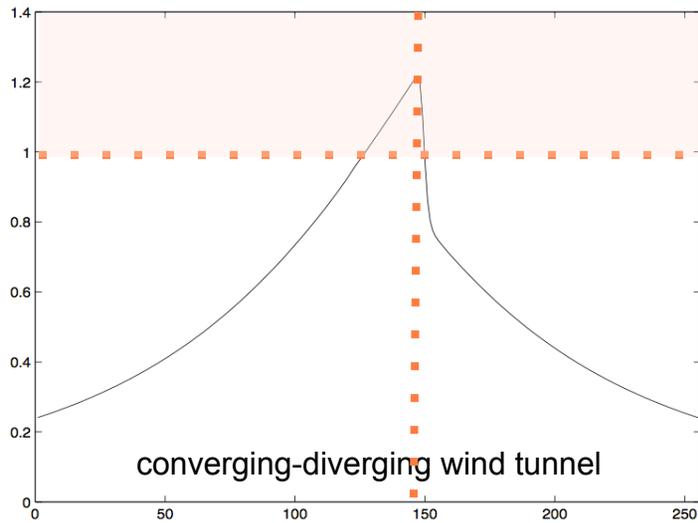


Figure 1: Mach distribution and the shock location.

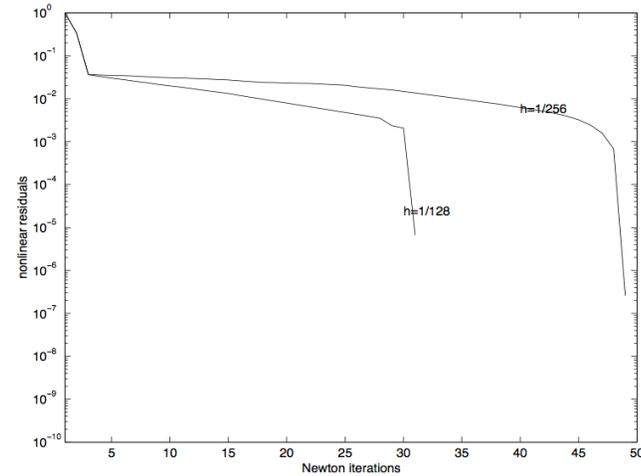


Figure 2: Nonlinear residual history of the inexact Newton's algorithm for the flow problem with mesh sizes  $h = 1/128$  and  $h = 1/256$ .

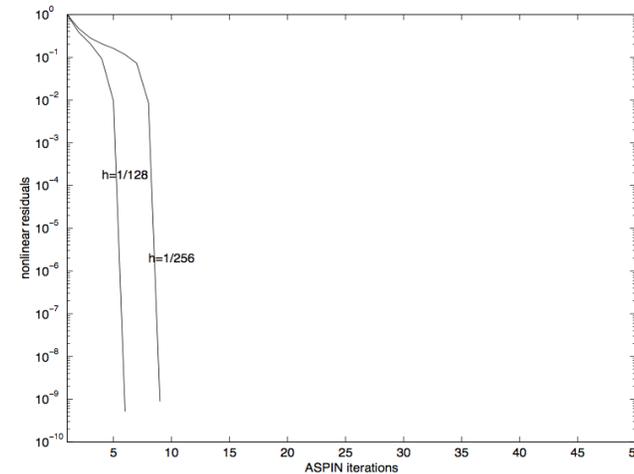


Figure 3: Nonlinear residual history of the additive Schwarz preconditioned inexact Newton's algorithm for the flow problem with mesh sizes  $h = 1/128$  and  $h = 1/256$ .

# Key idea

- **Newton method for a nonlinear system solves**  $F(u) = 0$ 
  - computes a global Jacobian matrix, and a global Newton step by solving the global linear system
- **Nonlinearly preconditioned Newton solves**  $\mathcal{F}(u) = 0$ 
  - implemented Jacobian-free through set of local problems on subsets of the *original* global nonlinear system
  - each of the linear systems for local Newton updates has only *local* scope and coordination
  - still global coordination in outer steps, hopefully *many fewer* than required in the original Newton method

## A selective history

[Lions, 1988] : *On the Schwarz Alternating Method. I*, 2-subdomain procedure for monotone nonlinear problems by alternating variational minimization in each subdomain

[Cai, Gropp, K & Tidriri, 1994] : *Newton-Krylov-Schwarz Methods in CFD*, a matrix-free method based on global linearization and local preconditioning

[Cai & Dryja, 1994] : *Domain decomposition methods for monotone nonlinear elliptic problems*, quadratic convergence proof for Newton, based on global linearization and local preconditioning

[Dryja & Hackbusch, 1997] : *On the nonlinear domain decomposition method*, an additive nonlinear Richardson iteration based on the solution of local nonlinear problems

[Cai & K, 2002] : *Nonlinearly preconditioned inexact Newton algorithms*, matrix-free Newton acceleration of [Dryja & Hackbusch, 1997]

# Equivalent system

- **Find solution  $u^*$  of  $F(u^*) = 0$  from  $\mathcal{F}(u^*) = 0$** 
  - using inexact Newton
  - linear systems solved with matrix-free Krylov
  - globalized with backtracking line search or trust region, etc.
- **$F(u) = 0$  and  $\mathcal{F}(u^*) = 0$  have the same solution**
- **$\mathcal{F}(w)$  is easily computable for  $w$  in  $R^n$**
- **$\mathcal{F}'(w)v$  is also easily computable for  $w, v$  in  $R^n$**

# Inexact Newton w/Backtracking

ALGORITHM 1 (INB).

An initial iterate  $x^{(0)}$  is given. For  $k = 0, 1, 2, \dots$  until convergence:

1. Find an inexact Newton direction  $d^{(k)}$  such that

$$(1.2) \quad \|F(x^{(k)}) - F'(x^{(k)})d^{(k)}\| \leq \eta_k \|F(x^{(k)})\|.$$

2. Determine a step size  $\lambda^{(k)}$  using a backtracking linesearch technique based on the function  $f(x) = \frac{1}{2}\|F(x)\|^2$ .
3. Compute a new approximate solution

$$(1.3) \quad x^{(k+1)} = x^{(k)} - \lambda^{(k)} d^{(k)}.$$

- **loose tolerance on forcing term  $\eta_k$  when INB used as an outer method**
- **tight tolerance when used as an inner method**
- **dependence on  $\eta_k$  studied later**

# Why nonlinear Schwarz preconditioning?

## **2000: Robustify Newton and improve its efficiency**

- Additive Schwarz Preconditioned Inexact Newton (ASPIN)
- interchange order of linearization and decomposition
- spend majority of effort on local problems
- local problems are smaller and better nonlinearly conditioned
- create better nonlinearly conditioned global problem, Jacobian-free
- high concurrency through domain decomposition

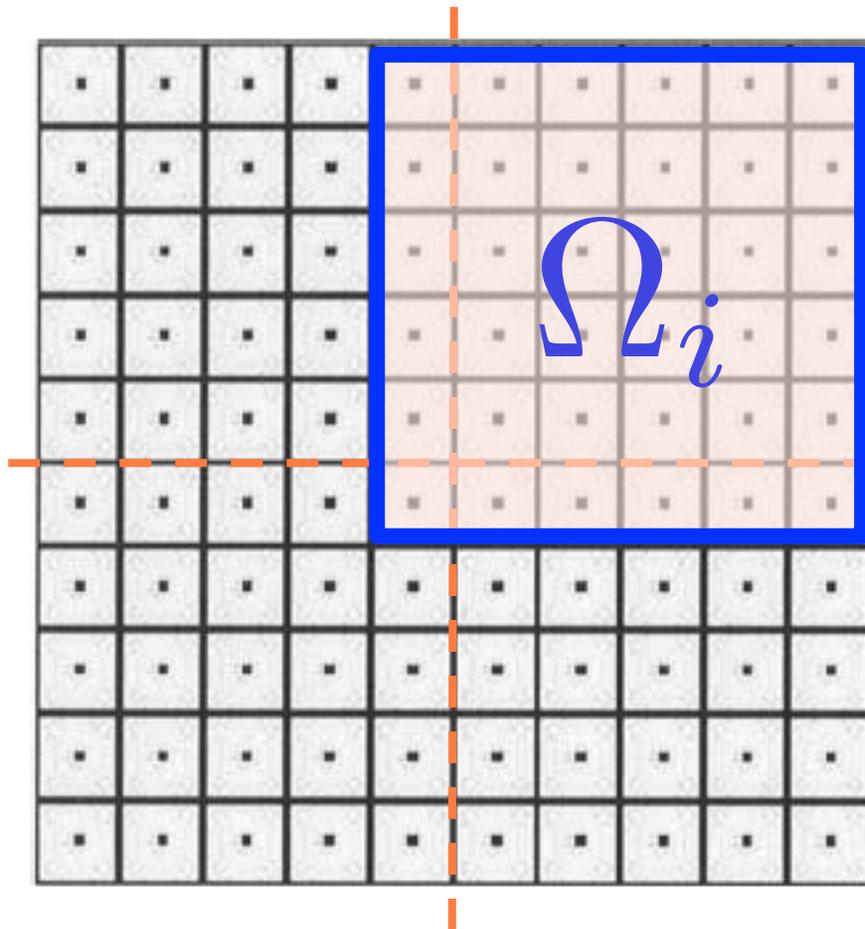
## **2010: Relax global synchronization requirements of Newton**

- fewer global synchronizations
- local synchronizations, asynchronous to each other

## **2015: Further robustify Newton for multicomponent systems**

- Multiplicative Schwarz Preconditioned Inexact Newton (MSPIN)
- precondition multiphysics through (sequential) uniphysics solves
- nest ASPIN (on subdomains) inside MSPIN, for reasons above

# ASPIN: nonlinear domain decomposition



$$\Omega = \bigcup_{i=1}^N \Omega_i, \quad i = 1, \dots, N$$

# ASPIN: construction through local solves

- **Concurrent (possibly overlapping) local solves for local corrections, using existing code**

$$F_{\Omega_i}(u - T_{\Omega_i}(u)) = 0, \quad i = 1, \dots, N$$

- **Sum for global residual**

$$\mathcal{F}(u) = \sum_{i=1}^N T_{\Omega_i}(u), \quad \bigcup_{i=1}^N \Omega_i = \Omega$$

- **Finite difference for global Jacobian-vector product**
- **No new code required for  $\mathcal{F}$  or its Jacobian  $\mathcal{J}$**

# ASPIN: 2-component example (nonoverlapping)

**Original system**

$$F(x) = 0, \quad x = \begin{bmatrix} u \\ v \end{bmatrix}, \quad F(x) = \begin{bmatrix} G(u, v) \\ H(u, v) \end{bmatrix}$$

**Transformed system**

$$\mathcal{F}(x) = \mathcal{F}(u, v) = \begin{bmatrix} g(u, v) \\ h(u, v) \end{bmatrix} = 0$$

**where  $(u, v)$  are obtained implicitly by solving independently**

$$G(u - g, v) = 0,$$

$$H(u, v - h) = 0.$$

## ASPIN: 2-component example (cont.)

**Jacobian of preconditioned system**

$$\mathcal{J}(u, v) = \begin{bmatrix} g_u & g_v \\ h_u & h_v \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial G}{\partial p}\right)^{-1} & \\ & \left(\frac{\partial H}{\partial q}\right)^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial u} & \frac{\partial H}{\partial q} \end{bmatrix}$$

**where**  $p = u - g(u, v)$  **and**  $q = v - h(u, v)$

**Since**  $(p, q)$  **approach**  $(u, v)$  **as the solution converges locally,**  
**the preconditioned Jacobian is locally well approximated by**  
**the readily computable**

$$\hat{\mathcal{J}}(u, v) = \begin{bmatrix} G_u^{-1} & \\ & H_v^{-1} \end{bmatrix} \begin{bmatrix} G_u & G_v \\ H_u & H_v \end{bmatrix} = \begin{bmatrix} G_u & \\ & H_v \end{bmatrix}^{-1} J(u, v)$$

**Diagonal blocks are identities, so linear conditioning depends**  
**on coupling strength in the off-diagonals**

## ASPIN: 2-component example (cont.)

Operationally, the approximate preconditioned matvec

$$\hat{\mathcal{J}}x = y$$

is straightforward, in terms of code for the original problem:

1. Perform the multiplication  $w = Jx$ ,  $w = [w_1, w_2]^T$ .
2. Solve  $G_u y_1 = w_1$  and  $H_v y_2 = w_2$ .
3. Form the result  $y = [y_1, y_2]^T$ .

**Generalization to 3 or more components is straightforward**

# Multiplicative generalizations

- [Kahou et al., 2007, 2008] : **multiplicative generalization of linear additive Schwarz (Richardson and Krylov-accelerated)**
  - applied to standard sparse test matrices
  - of limited interest due to lack of exploitation of concurrency
- [Ernst et al., 2007] : **multiplicative generalization of nonlinear additive Schwarz (Richardson)**
  - applied to acoustic-structure interaction (structure being nonlinear)
  - remarked: “inexact Newton generalization is future work”
- [Liu & Keyes, 2015] : **multiplicative Schwarz preconditioned inexact Newton (MSPIN)**
  - interesting for multicomponent problems, where the number of multiplicative stages is small
  - each stage represents a different component of the physics, for which an individual solver is presumed available

# Source of today's talk

SIAM J. SCI. COMPUT.  
Vol. 37, No. 3, pp. A1388–A1409

© 2015 Society for Industrial and Applied Mathematics

## FIELD-SPLIT PRECONDITIONED INEXACT NEWTON ALGORITHMS\*

LULU LIU<sup>†</sup> AND DAVID E. KEYES<sup>†</sup>

**Abstract.** The multiplicative Schwarz preconditioned inexact Newton (MSPIN) algorithm is presented as a complement to additive Schwarz preconditioned inexact Newton (ASPIN). At an algebraic level, ASPIN and MSPIN are variants of the same strategy to improve the convergence of systems with unbalanced nonlinearities; however, they have natural complementarity in practice. MSPIN is naturally based on partitioning of degrees of freedom in a nonlinear PDE system by field type rather than by subdomain, where a modest factor of concurrency can be sacrificed for physically motivated convergence robustness. ASPIN, originally introduced for decompositions into subdomains, is natural for high concurrency and reduction of global synchronization. We consider both types of inexact Newton algorithms in the field-split context, and we augment the classical convergence theory of ASPIN for the multiplicative case. Numerical experiments show that MSPIN can be significantly more robust than Newton methods based on global linearizations, and that MSPIN can be more robust than ASPIN and maintain fast convergence even for challenging problems, such as high Reynolds number Navier–Stokes equations.

**Key words.** nonlinear equations, nonlinear preconditioning, field splitting, Newton method, Navier–Stokes equations

**AMS subject classifications.** 65H10, 65H20, 65N22, 65N55

**DOI.** 10.1137/140970379

# MSPIN: 2-component example (nonoverlapping)

**Original system**

$$F(x) = 0, \quad x = \begin{bmatrix} u \\ v \end{bmatrix}, \quad F(x) = \begin{bmatrix} G(u, v) \\ H(u, v) \end{bmatrix}$$

**Transformed system**

$$\mathcal{F}(x) = \mathcal{F}(u, v) = \begin{bmatrix} g(u, v) \\ h(u, v) \end{bmatrix} = 0$$

**where  $(u, v)$  are obtained implicitly by solving *sequentially***

$$G(u - g, v) = 0,$$

$$H(u - g, v - h) = 0$$

# MSPIN: 2-component example (cont.)

## Jacobian of preconditioned system

$$\mathcal{J}(u, v) = \begin{bmatrix} \frac{\partial G}{\partial p} & \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}$$

where  $p = u - g(u, v)$  and  $q = v - h(u, v)$

As before, since  $(p, q)$  approaches  $(u, v)$  as the solution converges locally, the preconditioned Jacobian is locally well approximated by the readily computable

$$\hat{\mathcal{J}}(u, v) = \begin{bmatrix} G_p & \\ H_p & H_v \end{bmatrix}^{-1} \begin{bmatrix} G_p & G_v \\ H_p & H_v \end{bmatrix} = \begin{bmatrix} G_p & \\ H_p & H_v \end{bmatrix}^{-1} J(p, v)$$

## MSPIN: 2-component example (cont.)

Operationally, the approximate preconditioned matvec

$$\hat{\mathcal{J}}x = y$$

is again straightforward, in terms of code for the original problem:

1. Perform the multiplication  $w = Jx$ ,  $w = [w_1, w_2]^T$ .
2. Let  $z_2 = w_2$  and solve  $G_p z_1 = w_1$ .
3. Perform  $z_2 = z_2 - H_p z_1$  and set  $y_1 = z_1$ .
4. Solve  $H_v y_2 = z_2$ .
5. Form the result  $y = [y_1, y_2]^T$ .

**Generalization to 3 or more components is block triangular, as expected**

# Nonlinear preconditioning: theory

- **Assume original Jacobian  $J = F'(u)$  is continuous in a neighborhood  $D$  of the exact solution  $u^*$  and nonsingular at  $u^*$**
- **[Dryja & Hackbusch, 1997] : the original subproblems for  $T_{\Omega_i}$  are all uniquely solvable in a neighborhood of  $u^*$  in  $D$**
- **[Dryja & Hackbusch, 1997] : the matrix  $\Sigma_i(J_i^+)J$ , where  $J_i$  represents the Jacobian of the  $i^{\text{th}}$  subdomain extended to the full space, and  $J_i^+$  denotes its generalized inverse, is nonsingular in a neighborhood of  $u^*$  in  $D$**
- **Remark : if  $F(u) = b - Au$ , this is just the additive Schwarz preconditioned operator,  $\Sigma_i(A_i^+)A$**
- **The Jacobian of the ASPIN modified system  $\mathcal{J} = \mathcal{F}'(u)$  approaches  $\Sigma_i(J_i^+)J$  as  $u$  approaches  $u^*$**

# Nonlinear preconditioning: theory

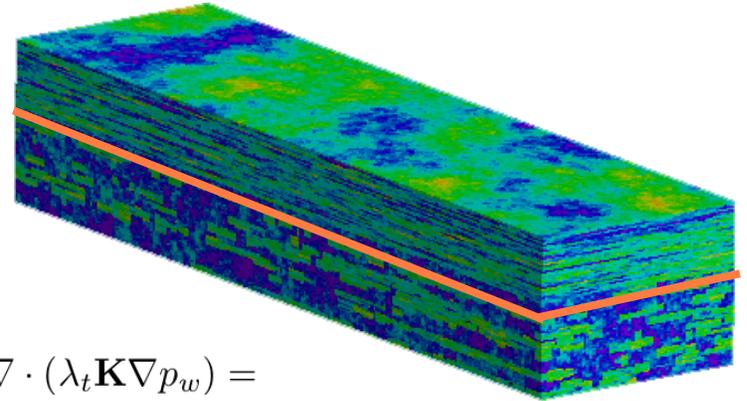
- [Cai & K, 2002] :  $F(u)$  and ASPIN's  $\mathcal{F}(u)$  are equivalent in that they possess the same solution in a neighborhood of  $D$
- [An, 2005] : ASPIN local convergence guaranteed
  - superlinear if forcing term in inexact Newton approaches 0
  - quadratic if forcing term approaches 0 like  $O(\|\mathcal{F}(\bullet)\|)$
- [Liu & K, 2014] :  $F(u)$  and MSPIN's  $\mathcal{F}(u)$  are equivalent in that they possess the same solution in a neighborhood of  $D$
- [Liu & K, 2015] : MSPIN local convergence guaranteed
  - superlinear if forcing term in inexact Newton approaches 0
  - quadratic if forcing term approaches 0 like  $O(\|\mathcal{F}(\bullet)\|)$

# Examples

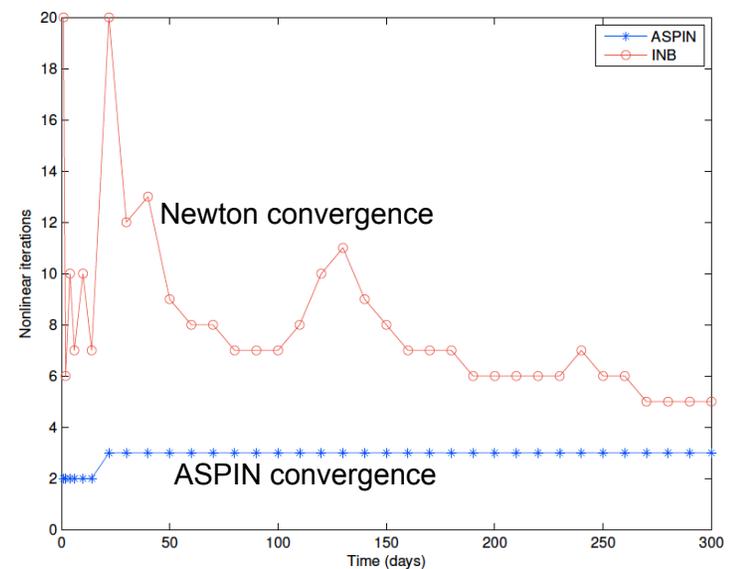
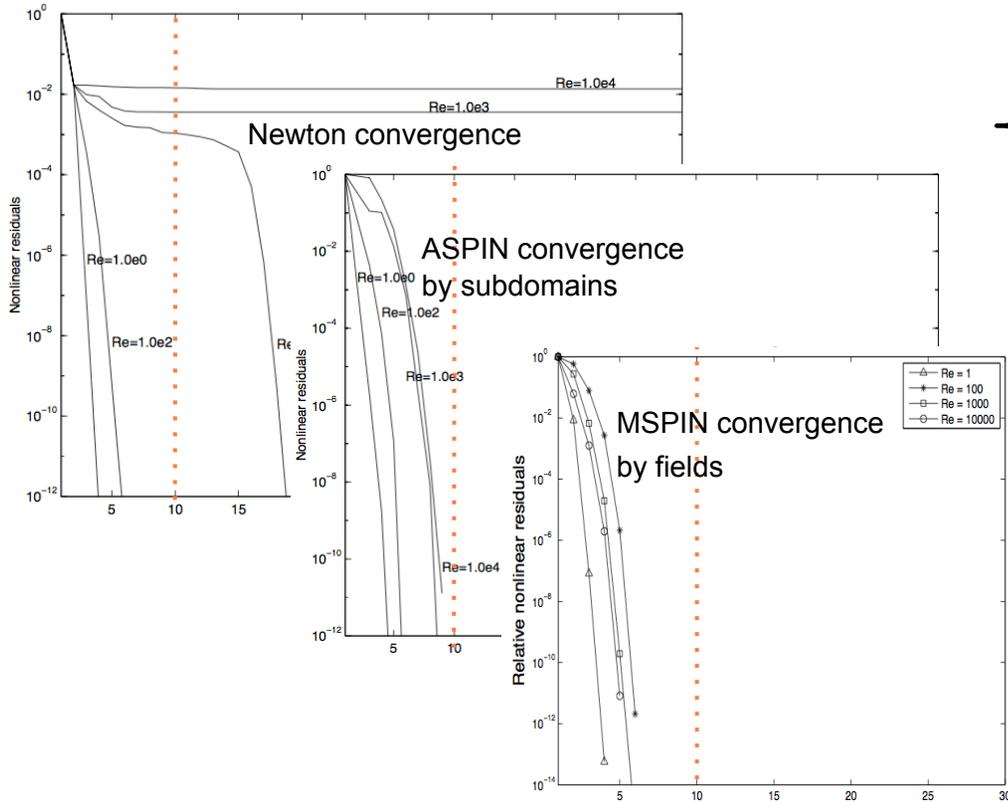
Driven cavity model (ex19 in PETSc)

$$\begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \\ -\frac{1}{Re} \Delta \omega + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = 0. \end{cases}$$

reservoir model (SPE10)



$$\begin{cases} -\nabla \cdot (\lambda_t \mathbf{K} \nabla p_w) = \\ q_t + \nabla \cdot (\lambda_n \mathbf{K} \nabla p_c) - \nabla \cdot ((\lambda_n \rho_n + \lambda_w \rho_w) \mathbf{K} \mathbf{g}), \\ \phi \frac{\partial S_w}{\partial t} - \nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})) = q_w. \end{cases}$$



# ASPIN: PETSc implementation

petsc-3.6.0 2015-06-09

[Report Typos and Errors](#)

## SNESASPIN

Helper [SNES](#) type for Additive-Schwarz Preconditioned Inexact Newton

### Options Database

- npc\_snes\_ - options prefix of the nonlinear subdomain solver (must be of type NASM)
- npc\_sub\_snes\_ - options prefix of the subdomain nonlinear solves
- npc\_sub\_ksp\_ - options prefix of the subdomain Krylov solver
- npc\_sub\_pc\_ - options prefix of the subdomain preconditioner

Notes: This routine sets up an instance of NETWORKLS with nonlinear left preconditioning. It differs from other

**similar functionality in SNES as it creates a linear shell matrix that corresponds to the product**

$$\sum_{i=0}^{N_b} J_b(\{X^b_{\text{converged}}\})^{-1} J(X + \sum_{i=0}^{N_b} (X^b_{\text{converged}} - X^b))$$

which is the ASPIN preconditioned matrix. Similar solvers may be constructed by having matrix-free differencing of nonlinear solves per linear iteration, but this is far more efficient when subdomain sparse-direct preconditioner factorizations are reused on each application of  $J_b^{-1}$ .

### See Also

[SNESCreate\(\)](#), [SNES](#), [SNESSetType\(\)](#), [SNESNEWTONLS](#), [SNESNASM](#), [SNESGetNPC\(\)](#), [SNESGetNPCSide\(\)](#)

**Level:**intermediate

**Location:**[src/snes/impls/nasm/aspin.c](#)

## 2-unknown algebraic example [Hwang, 2004]

For ease of manipulation and visualization, consider

$$\begin{aligned}F_1(x_1, x_2) &= (x_1 - x_2^3 + 1)^3 - x_2^3, \\F_2(x_1, x_2) &= 2x_1 + 3x_2 - 5.\end{aligned}$$

**For ASPIN (Jacobi-like)**

$$F_1(x_1 - \delta_1^J, x_2) = 0$$

$$F_2(x_1, x_2 - \delta_2^J) = 0$$



$$\begin{aligned}\delta_1^J(x_1, x_2) &= x_1 - x_2^3 + 1 - x_2, \\ \delta_2^J(x_1, x_2) &= \frac{2}{3}x_1 + x_2 - \frac{5}{3}.\end{aligned}$$

**For MSPIN (Gauss-Seidel-like)**

$$F_1(x_1 - \delta_1^{GS}, x_2) = 0$$

$$F_2(x_1 - \delta_1^*, x_2 - \delta_2^{GS}) = 0$$



$$\begin{aligned}\delta_1^{GS}(x_1, x_2) &= x_1 - x_2^3 + 1 - x_2, \\ \delta_2^{GS}(x_1, x_2) &= \frac{2}{3}x_2^3 + \frac{5}{3}x_2 - \frac{7}{3}.\end{aligned}$$

# Original vs. ASPIN vs. MSPIN

$$F_1(x_1, x_2) = (x_1 - x_2^3 + 1)^3 - x_2^3,$$
$$F_2(x_1, x_2) = 2x_1 + 3x_2 - 5.$$

**One ninth-order, one linear, both equations couple unknowns**

$$\delta_1^J(x_1, x_2) = x_1 - x_2^3 + 1 - x_2,$$
$$\delta_2^J(x_1, x_2) = \frac{2}{3}x_1 + x_2 - \frac{5}{3}.$$

**All have  
same root,  
namely  
(1,1)**

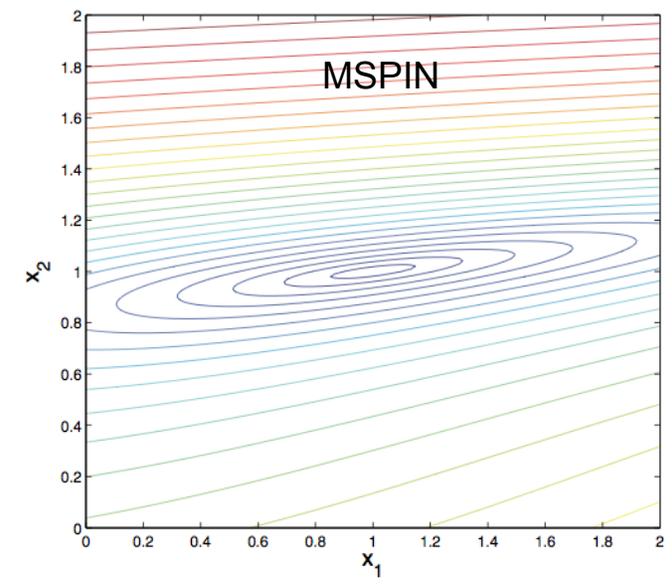
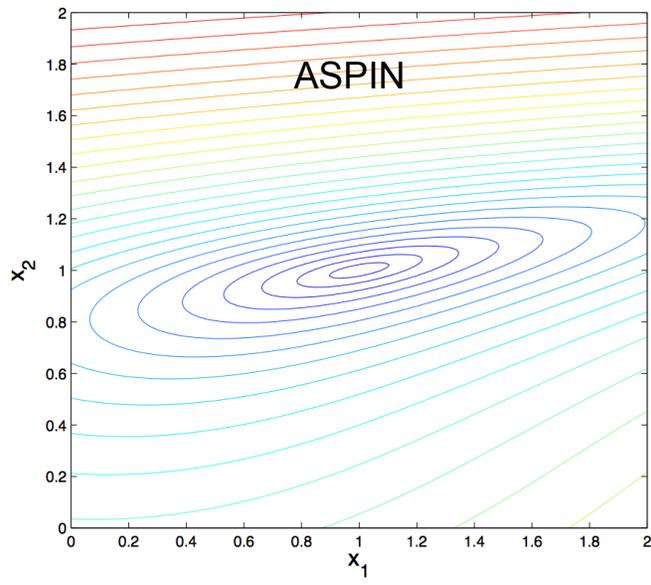
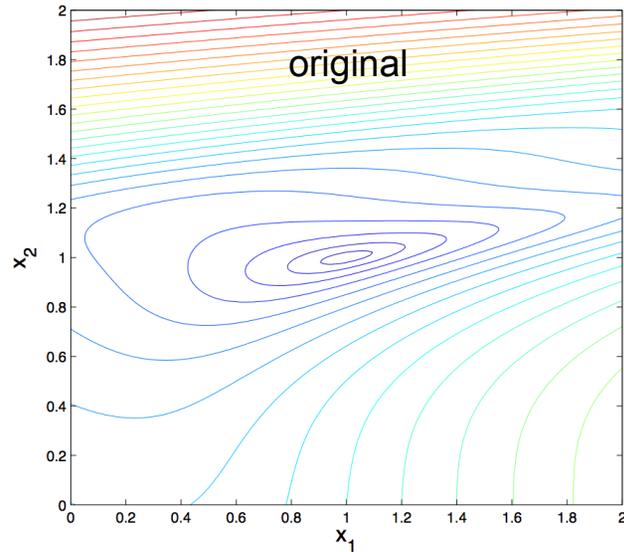
**One third-order, one linear, both equations couple unknowns**

$$\delta_1^{GS}(x_1, x_2) = x_1 - x_2^3 + 1 - x_2,$$
$$\delta_2^{GS}(x_1, x_2) = \frac{2}{3}x_2^3 + \frac{5}{3}x_2 - \frac{7}{3}.$$

**Both third-order, one equation decouples**

# Original vs. ASPIN vs. MSPIN

Contours of  $\log( \|F(x_1, x_2)\| + 1 )$



# Original vs. ASPIN vs. MSPIN

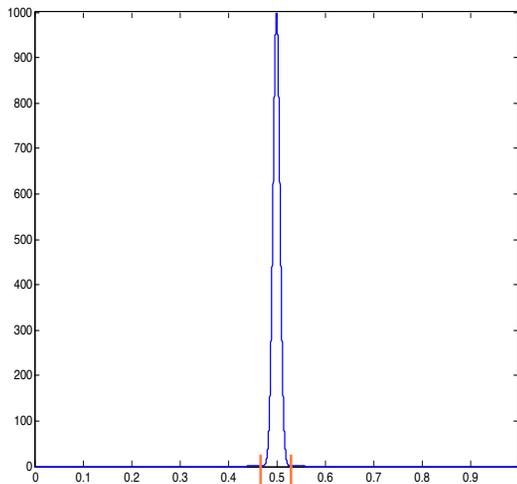
TABLE 1

*The number of nonlinear iterations. The outer global tolerance is  $10^{-8}$ , and the inner component tolerances are both  $10^{-3}$ .*

Initial guess $x_0$	INB	ASPIN	MSPIN
$x_0 = (0, 0)^T$	11	7	6
$x_0 = (0, 2)^T$	10	7	5
$x_0 = (2, 0)^T$	1	8	6
$x_0 = (2, 2)^T$	11	7	5

# 1D BVP example

[Lanzkron, Rose & Wilkes, 1997]



$$-u'' + u^3 + (4 \times 10^8(x - 0.5)^2 - 2 \times 10^4)u - 10^9 e^{-3\left(\frac{x-0.5}{0.01}\right)^2} = 0$$

$$x \in (0, 1), \quad u(0) = 0, \quad u(1) = 0$$

$$u(x) = 10^3 e^{-\left(\frac{x-0.5}{0.01}\right)^2}$$

*Comparison of the number of nonlinear iterations for different methods in the rightmost three columns. “No. points” indicates the number of grid points used to discretize the ODE. The points between  $G_{\text{left}}$  and  $G_{\text{right}}$  are unknowns in  $G$ .*

No. points	$G_{\text{left}}$	$G_{\text{right}}$	Its. INB	Its. ASPIN	Its. MSPIN
100	48	53	5	2	1
500	236	265	5	2	2
1000	471	530	6	2	1
5000	2351	2650	5	2	2

# DAE example

[PETSc, ex28]

$$(5.4) \quad -(vu_x)_x + \lambda u^2 = 1 \text{ on } (0, 1), \quad \text{subject to } u(0) = 0, \quad u(1) = 1,$$

$$(5.5) \quad \exp(v - 1) + v = \frac{1}{\frac{1}{1+u} + \frac{1}{1+u_x^2}},$$

*Global nonlinear and linear iterations using globalized INB, ASPIN and MSPIN.  $\epsilon_{\text{global-nonlinear-rtol}} = 10^{-10}$ ,  $\epsilon_{\text{global-linear-rtol}} = 10^{-8}$ ,  $\epsilon_{\text{sub-nonlinear-rtol}} = 10^{-3}$ . “\*” indicates that linear iterations are not available, since the nonlinear methods stagnate at the line search.*

Methods	Number of PIN iterations				
	$\lambda = 0$	$\lambda = 100$	$\lambda = 1000$	$\lambda = 3000$	$\lambda = 5000$
INB	6	6	15	-	9
ASPIN	6	-	-	-	-
MSPIN	5	5	5	5	5
Average number of GMRES iterations per PIN					
INB	12	18	14	*	7
ASPIN	21	*	*	*	*
MSPIN	10	9	8	8	7

# 3-field PDE example

[PETSc, ex19]

$$(5.6) \quad \begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \\ -\frac{1}{Re} \Delta \omega + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = 0. \end{cases}$$

Here  $u = 1, v = 0$  on the top boundary and  $u = 0, v = 0$  on the other boundaries. The boundary condition on  $\omega$  is given by its definition:

$$(5.7) \quad \omega(x, y) = -\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}.$$

---

**MSPIN  
splitting**

$$G : \begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \end{cases}$$

**$G, H$  systems  
are both linear  
among their  
own unknowns**

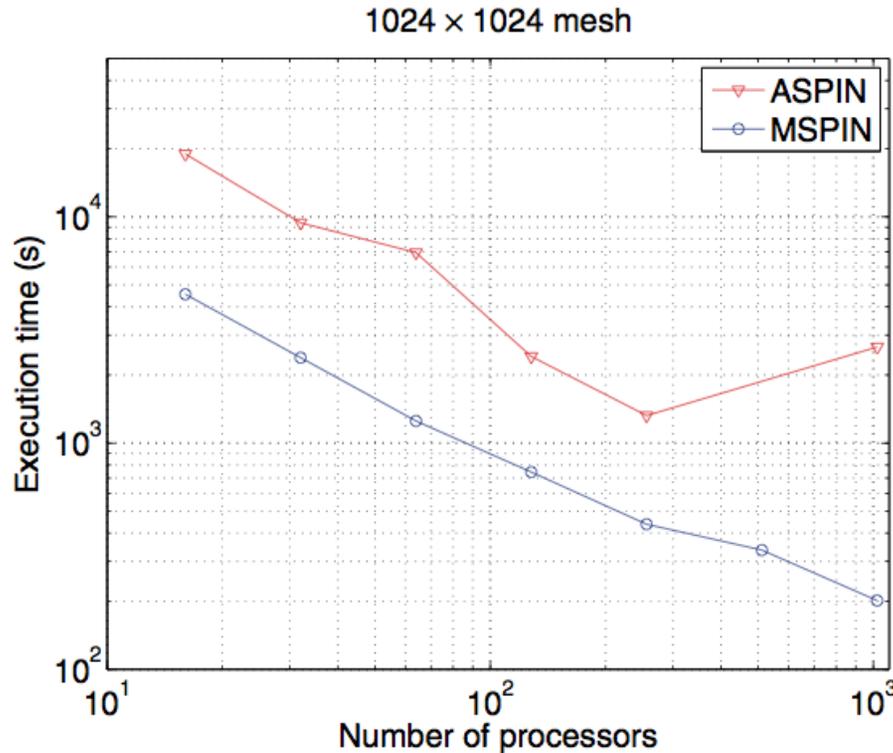
$$H : -\frac{1}{Re} \Delta \omega + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = 0.$$

# MSPIN: 3-field PDE example

Global nonlinear and linear iterations using globalized INB, ASPIN, and MSPIN on different mesh sizes. The initial guess is zero for  $u$ ,  $v$ , and  $\omega$ .  $\epsilon_{\text{global-linear-rtol}} = 10^{-6}$ ,  $\epsilon_{\text{global-nonlinear-rtol}} = 10^{-10}$ . The finite difference step size for the matrix-free Jacobian applications is  $10^{-8}$ . “\*\*” indicates that nonlinear iterations are not available, because linear iterations fail to converge after 10,000 steps.

64 × 64 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	17
ASPIN	4	9	10
MSPIN	4	5	4
Average number of GMRES iterations per PIN			
INB	19	24	26
ASPIN	23	37	39
MSPIN	13	18	15
128 × 128 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	**
ASPIN	4	9	13
MSPIN	4	5	5
Average number of GMRES iterations per PIN			
INB	37	62	-
ASPIN	24	42	14
MSPIN	14	21	20
256 × 256 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	**
ASPIN	4	10	18
MSPIN	4	6	6
Average number of GMRES iterations per PIN			
INB	93	200	-
ASPIN	25	47	149
MSPIN	15	24	28

# MSPIN: 3-field PDE example



**Linear  
subsystems  
solved with  
hypre's  
BoomerAMG**

FIG. 5. Strong scaling for the driven cavity flow problem on a  $1024 \times 1024$  mesh at Reynolds number 1000. The initial guess is still zero for  $u, v, \omega$ .  $\epsilon_{\text{global-linear-rtol}} = 10^{-3}$ ,  $\epsilon_{\text{global-nonlinear-rtol}} = 10^{-8}$ ,  $\epsilon_{\text{sub-rtol}} = 10^{-3}$ , and  $\epsilon_{\text{Jac-rtol}} = 10^{-3}$ .  $\epsilon_{\text{sub-rtol}}$  denotes the relative tolerance for the subproblems (which are linear in this example), and we specify  $\epsilon_{\text{Jac-rtol}}$  as the relative tolerance for the linear problems in (2.13) and (2.29). The finite difference step size for the matrix-free Jacobian applications is  $10^{-8}$ . Execution time for ASPIN using 512 processors is not shown since it fails to converge on this mesh and this Reynolds number from a zero initial guess.

# MSPIN: 3-field PDE example

Execution times for strong scaling of the lid-driven cavity for ASPIN and MSPIN on a  $256 \times 256$  mesh at Reynolds number 1000, for tight and loose relative convergence tolerances on the subproblems and global preconditioner linear systems solutions. The initial guess is zero for  $u$ ,  $v$ , and  $\omega$ .  $\epsilon_{\text{global-linear-rtol}} = 10^{-6}$ ,  $\epsilon_{\text{global-nonlinear-rtol}} = 10^{-8}$ .  $\epsilon_{\text{sub-rtol}}$  denotes the relative tolerance for the subproblems (which are linear in this example), and we specify  $\epsilon_{\text{Jac-rtol}}$  as the relative tolerance for the linear problems in (2.13) and (2.29). The finite difference step size for the matrix-free Jacobian applications is  $10^{-8}$ . “ $N_p$ ” indicates the number of processors, which does not have to be square. Performance for INB is not shown since it fails to converge on this mesh and this Reynolds number from a zero initial guess.

Execution time (s)					
256 × 256 mesh					
Methods	$N_p$	$\epsilon_{\text{sub-rtol}} = 10^{-3}$	$\epsilon_{\text{sub-rtol}} = 10^{-3}$	$\epsilon_{\text{sub-rtol}} = 10^{-6}$	$\epsilon_{\text{sub-rtol}} = 10^{-6}$
		$\epsilon_{\text{Jac-rtol}} = 10^{-3}$	$\epsilon_{\text{Jac-rtol}} = 10^{-6}$	$\epsilon_{\text{Jac-rtol}} = 10^{-3}$	$\epsilon_{\text{Jac-rtol}} = 10^{-6}$
ASPIN	4	2363.98	3273.88	2194.43	3219.34
	16	687.68	943.91	654.32	976.95
	32	397.12	589.86	395.26	588.95
	64	272.4	412.27	276.03	405.92
MSPIN	4	175.31	245.59	199.64	248.18
	16	57.04	74.06	56.74	74.48
	32	32.17	45.86	34.55	46.33
	64	22.31	31.64	23.90	31.79

# Comment

- **Newton was not doomed before nonlinear preconditioning**
- **Other relevant globalization methods for the driven cavity**
  - **mesh continuation**: approach the problem on the mesh of desired resolution by initial guesses recursively built up from easier Newton problems on coarser meshes
  - **Reynolds number continuation**: approach the problem at the desired Reynolds by initial guesses projected by Davidenko's method from the easier low Reynolds limit
  - **pseudo-transient continuation**: approach the steady state by a transient approach in the vorticity equation, with implicit time step eventually approaching infinity
- **These may also be combined with nonlinear preconditioning**

# Caveat

- **Newton is ever more art than science...**
- **Picking  $u, v, \dots$  and corresponding  $G, H, \dots$  is not trivial**
- **Different groupings and different orderings can change the quality of the preconditioning – even dramatically**
  - **generally good to order the linear subsystem (or the “least” nonlinear subsystems) first**
  - **generally good to try to keep the “most” nonlinear subsystems as small as possible and order last**
  - **sometimes splitting the systems by component yields equations that are linear among their own local unknowns**

# Future work

- **Applications: gain more experience on problems difficult to converge globally any other way**
- **Improvements: the modified Jacobian is known only in the form of matvecs and is therefore a challenge to precondition further**
  - should inner preconditioning therefore include a spatially hierarchical component?
- **Theory: try to come up with measures of nonlinearity for different subsystems**
  - will vary with arguments
  - cannot deduce from form of equations alone
- **Software: create high performance implementations**
  - exploit the permitted asynchrony of inner Newton subproblems, with work-stealing
  - nest domain-split ASPIN inside of field-split MSPIN

**Thank you!**