B-spline Adaptive Collocation/Runge-Kutta Software with Interpolation-based Spatial Error Estimation for the Error Controlled Numerical Solution of PDEs

Paul Muir
Saint Mary's University
muir@smu.ca

Thanks: ANM for PDEs w Applc organizing team,
NSERC, MITACS, ACEnet, AARMS, Saint Mary's University

Preamble

- **Production Level** software for adaptive **error controlled** numerical solution of 1D PDEs
- Intro to next talk: "eBACOLI: a time- and space-adaptive multi-scale PDE solver" by Ray Spiteri
- Applications in epidemiology, electro-physiology, etc.

Outline

- **Error Control Software**
- Setting the Context: (On-going work)
  B-spline Gaussian Collocation Error Control Software for 2D PDEs
- 1D PDE Software:
  B-spline Adaptive Collocation (BACOL) Error Control Software Family
- **BACOLRI** (New)

Error Control Software

- **Error Control Software:**
  Implements adaptive computation to obtain approximate solution such that corresponding **error estimate for approximate solution satisfies user tolerance**
- **Two important advantages:**
  - User can have reasonable confidence that numerical solution has **error consistent with tolerance**
  - Cost of **computation will be consistent with requested accuracy**

Setting the Context

- 2D Burgers Equation [Velivelli, Bryden, 2006]

$$u_t = \epsilon u_{xx} + \epsilon u_{yy} - u u_x - u u_y, (x, y) \in (0, 1) \times (0, 1), t > 0,$$

- Boundary and initial conditions chosen from exact solution

$$u(x, y, t) = \frac{1}{1 + e^{\left(\frac{x+y-t}{2\epsilon}\right)}}$$

Setting the Context

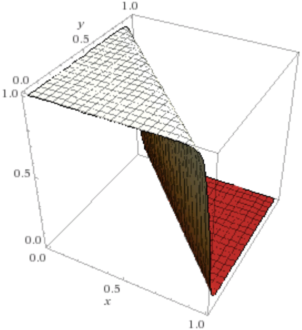


Figure: Solution of 2D Burgers Equation, $\epsilon = 10^{-2}$

Setting the Context

B-spline Gaussian Collocation for 2D PDEs [Li, Muir 2013]

- Rectangular grid based on meshes of $N+1$ and $M+1$ points, partitioning $[a,b]$ and $[c,d]$ respectively.
- Piecewise polynomials of degree p in x and y represented in terms of B-spline bases: $\{B_i(x)\}_{i=1}^{NC}$ and $\{D_i(y)\}_{i=1}^{MC}$, $NC = N(p-1)+2$, $MC = M(p-1)+2$
- Approximate solution, $\mathbf{U}(x,y,t)$:

$$\mathbf{U}(x,y,t) = \sum_{i=1}^{NC}\sum_{j=1}^{MC} \mathbf{w}_{ij}(t) B_i(x) D_j(y)$$

with unknown vector time-dependent coefficients, $\mathbf{w}_{ij}(t)$

Setting the Context

- $\mathbf{U}(x, y, t)$, for a given t, **required to satisfy PDE at collocation points** $\{\xi_i\}_{i=2}^{NC-1}$ in x and $\{\gamma_j\}_{j=2}^{MC-1}$ in y, respectively, where ξ_i, γ_j are **images of** $(p-1)$**-point Gaussian quadrature rule** mapped onto each subinterval of x and y meshes, respectively
- $\mathbf{U}(x, y, t)$ also required to satisfy boundary conditions at images of Gauss points on each boundary subinterval in x and y domains
- Resultant system of **differential-algebraic equations** (DAEs) solved using DAE solver, DASPK [Brown, Hindmarsh, Petzold 1994], designed for large sparse DAE systems
- $\Rightarrow \mathbf{w}_{ij}(t)$, from which $\mathbf{U}(x, y, t)$ can be constructed

Setting the Context

- **Temporal error controlled computation** of B-spline coefficients by DASPK
- $\mathbf{U}(x, y, t)$ has spatial error that is $O(h^{p+1})$, where h is the spatial mesh spacing ($\Delta x \approx \Delta y \approx h$)
- But software does not have **spatial adaptivity, spatial error estimation, or control of spatial error estimate** (Future work)

Setting the Context

Development of high quality error control B-spline Gaussian collocation software for 2D PDEs depends on . . .

B-spline Gaussian Collocation software for 1D PDEs:

- BACOL [Wang, Keast, Muir 2004a,b,c]
- BACOLR [Wang, Keast, Muir 2008]
- BACOLI [Arsenault, Smith, Muir 2009], [Arsenault, Smith, Muir, Pew 2012], [Muir, Pew 2015], [Pew, Li, Muir 2016]
- Performance Analysis of BACOL, BACOLR, BACOLI [Pew, Li, Tannahill, Muir, Fairweather 2018]
- **(New) BACOLRI [Pew, Tannahill, Murtha, Muir 2018]**

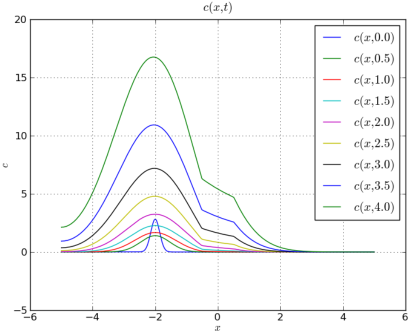
1D PDE Example: Brain Tumor Invasion Model (BTIM)

- [Papadomanolaki, Saridakis, 2009],
  Scaled tumor concentration $u(x,t)$

$$u_t(x,t) = (D(x)u_x(x,t))_x, \quad x \in [a,b], \quad t \geq 0$$

- Boundary conditions $u_x(a,t) = 0, u_x(b,t) = 0$
- Initial condition $u(x,0) = f(x)$, where $f(x)$ represents initial tumor concentration
- $D(x)$, scaled diffusion coefficient, equal to 1 in white matter region and γ in grey matter region
- We approximate this step function by
  $D_l(x) = \left(\left(\frac{1}{e^{-l(x-w_1)}+1}\right) + \left(\frac{1}{e^{l(x-w_2)}+1}\right) - 1\right)(1-\gamma) + \gamma$

BTIM Solution $c(x,t) = e^t \cdot u(x,t)$



Software for 1D PDEs: B-spline Gaussian Collocation

BACOL family of error control 1D PDE solvers:

- Mesh of $N+1$ points, partitioning spatial domain $[a,b]$
- Approximate solution $\mathbf{U}(x,t)$:

$$\mathbf{U}(x,t) = \sum_{i=1}^{NC} \mathbf{w}_i(t) B_i(x),$$

  $B_i(x)$, B-spline basis functions, degree p, $NC = N(p-1)+2$, unknown vector time-dependent coefficients, $\mathbf{w}_i(t)$
- $\mathbf{U}(x,t)$, for given t, required to satisfy boundary conditions and PDE at images of $(p-1)$ Gauss points on each subinterval

Software for 1D PDEs: B-spline Gaussian Collocation

- Resultant system of DAEs solved using DASSL [Brennan, Campbell, Petzold 1995] (Family of Backward Differentiation Formulas (BDFs), orders 1 through 5 (hp adaptivity in time)) or RADAU5 (IRK method of order 5, (h adaptivity in time)) [Hairer, Wanner 1995]
- **Temporal error controlled computation of $\mathbf{w}_i(t)$**, from which $\mathbf{U}(x,t)$ can be constructed
- $\mathbf{U}(x,t)$ has spatial error that is $O(h^{p+1})$, where h is the spatial mesh spacing
- BACOL/BACOLR allow $p \in \{2, \ldots, 10\} \Rightarrow$ orders $3, \ldots, 11$

Software for 1D PDEs: B-spline Gaussian Collocation

- **BACOL** [Wang, Keast, Muir, 2004a]: high order temporal error estimates computed and controlled by DASSL
- **BACOLR** [Wang, Keast, Muir, 2008]: DASSL replaced by RADAU5, **better stability**, relevant for certain classes of problems, e.g., Schrödinger equation, where BDFs have stability issues; (BDFs have stability issues for problems which lead to Jacobian matrices with eigenvalues near imaginary axis); **BACOL fails unless restricted to first/second order in time**
- BACOL, BACOLR: **spatial error estimates obtained by computing a second higher order approximate solution,** $\bar{\mathbf{U}}(x,t)$, using B-splines of degree $p+1$

Software for 1D PDEs: B-spline Gaussian Collocation

Spatial error control:

- After each accepted time step, spatial error estimate is computed and compared with user tolerance
- If tolerance not satisfied $\Rightarrow$ **adaptive spatial mesh refinement based on direct equidistribution of spatial error estimates** (Use of equidistribution equation directly, no moving mesh PDE)
- Remeshing involves **high-order interpolation** of solution info from previous spatial mesh to new spatial mesh
- Allows code to continue in **"warm start"** mode

Software for 1D PDEs: B-spline Gaussian Collocation

- BACOL shown to have superior performance to similar packages [Wang, Keast, Muir, 2004c], especially for problems with sharp moving layers and sharp tolerance requests
- BACOLR shown to have comparable performance to BACOL on standard problems and much superior performance for problems where stability of BDFs is an issue
- **However, for either code, computation of $\bar{\mathrm{U}}(x,t)$ for spatial error estimate essentially doubles cost of computation**

Software for 1D PDEs: B-spline Gaussian Collocation

BACOLI [Pew, Li, Muir, 2016]: New interpolation-based spatial error estimation/control schemes

- Computation of $\bar{\mathbf{U}}(x,t)$ is removed and replaced with computation of one of two **piecewise polynomial interpolants** to $\mathbf{U}(\mathbf{x},\mathbf{t})$
- **Superconvergent Interpolant** (SCI) scheme [Arsenault, Smith, Muir, 2009]; Hermite-Birkhoff interpolant based on evaluation of $\mathbf{U}(x,t)$ at points where it is superconvergent
- SCI is one spatial order higher than $\mathbf{U}(x,t)$

Software for 1D PDEs: B-spline Gaussian Collocation

- **Lower Order Interpolant** (LOI) scheme [Arsenault, Smith, Muir, Pew, 2013]); Hermite-Birkhoff interpolant based on evaluation of $\mathbf{U}(x,t)$ at certain points so that interpolation error agrees asymptotically with error of a collocation solution of one order lower
- LOI is one spatial order lower than $\mathbf{U}(x,t)$
- BACOLI shown in [PLM 2016] to be about twice as fast as BACOL

Software for 1D PDEs: B-spline Gaussian Collocation

Two spatial error control modes:

- Superconvergent Interpolant (SCI) scheme: estimates and controls error estimate for $\mathbf{U}(x,t) \Rightarrow$ **standard (ST) spatial error control**
- Lower Order Interpolant (LOI) scheme: estimates and controls error estimate for collocation solution of one order lower $\Rightarrow$ **Local Extrapolation (LE) spatial error control** (Runge-Kutta formula pairs)
- BACOL/BACOLR return $\mathbf{U}(x,t)$ and estimate and control error for $\mathbf{U}(x,t) \Rightarrow$ **ST error control**
- With a slight modification, BACOL/BACOLR could return $\bar{\mathbf{U}}(x,t)$; however, spatial error estimate/control would still be for $\mathbf{U}(x,t) \Rightarrow$ **LE error control**

Software for 1D PDEs: B-spline Gaussian Collocation

BACOLI: Available software:

- **Collected algorithms of the ACM,** http://calgo.acm.org/, Algorithm 962
- **BACOLI77:** cs.smu.ca/∼muir/BACOLI-3_Webpage.htm; Fortran 77
- **BACOLI95:** cs.smu.ca/∼muir/BACOLI-3_Webpage.htm; Fortran 95; Easier to use: no work arrays, many optional parameters
- Python interface **BACOLI_PY** (in development)
- Production level software:
  - General problem class - not problem specific;
  - Used by "arms length" users; avoid user chosen parameters;
  - Requires extensive testing and performance analysis;
  - Documentation; user guide; website; on-going maintenance

Software for 1D PDEs: B-spline Gaussian Collocation

The BACOL family of error control B-spline Gaussian collocation software packages for 1D PDEs

$$
\begin{array}{ccccc}
\textbf{BACOL} & \Rightarrow & \textit{Stability} & \Rightarrow & \textbf{BACOLR} \\
\Downarrow & & & & \Downarrow \\
\textit{Error} & & & & \textit{Error} \\
\textit{Estimate} & & & & \textit{Estimate} \\
\Downarrow & & & & \Downarrow \\
\textbf{BACOLI} & \Rightarrow & \textit{Stability} & \Rightarrow & \textbf{BACOLRI}
\end{array}
$$

Software for 1D PDEs: B-spline Gaussian Collocation

New Software: **BACOLRI**:

- Major modification of BACOLR to improve efficiency of spatial error estimation
- Second approximate solution, $\bar{\mathbf{U}}(x,t)$, is removed from BACOLR
- Implementation of SCI and LOI schemes within BACOLR
- Twice as efficient as BACOLR
- Comparable to BACOLI on standard problems but
- Able to handle, e.g., Schrödinger equations, where stability of BDFs is an issue $\Rightarrow$ BACOLI will fail unless restricted to first/second order in time - which makes it much slower

Software for 1D PDEs: B-spline Gaussian Collocation

- The Two Layer Burgers Equation (TLBE)
  - $u_t = -uu_x + \epsilon u_{xx}, \qquad 0 < x < 1, \quad t > 0$
  - Initial and boundary conditions taken from the exact solution

$$u(x,t) = \frac{0.1e^{-A} + 0.5e^{-B} + e^{-C}}{e^{-A} + e^{-B} + e^{-C}},$$

    where $A = \frac{0.05}{\epsilon}(x - 0.5 + 4.95t)$, $B = \frac{0.25}{\epsilon}(x - 0.5 + 0.75t)$, $C = \frac{0.5}{\epsilon}(x - 0.375)$, and ϵ is a problem dependent parameter
  - We set $\epsilon = 10^{-4}$

Software for 1D PDEs: B-spline Gaussian Collocation

- **BACOLR vs. BACOLRI**
- TLBE $\epsilon = 10^{-4}$, degree of B-spline basis functions $p = 6$, $tol = 10^{-2}, \ldots, 10^{-10}$, compare:
  - BACOLR/ST (BACOLR using Standard (ST) spatial error control)
  - BACOLR/LE (BACOLR using Local Extrapolation (LE) spatial error control)
  - BACOLRI/ST (BACOLRI using ST spatial error control)
  - BACOLRI/LE (BACOLRI using LE spatial error control)

Software for 1D PDEs: B-spline Gaussian Collocation

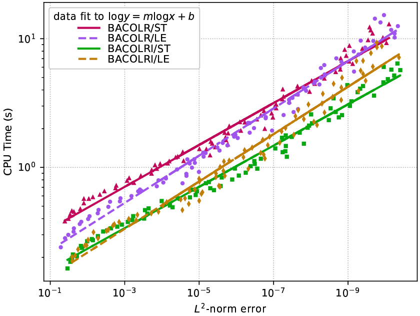


Figure: Error vs. CPU Time for BACOLR/ST, BACOLR/LE, BACOLRI/ST, and BACOLRI/LE, TLBE $\epsilon = 10^{-4}$, $p = 6$

Software for 1D PDEs: B-spline Gaussian Collocation

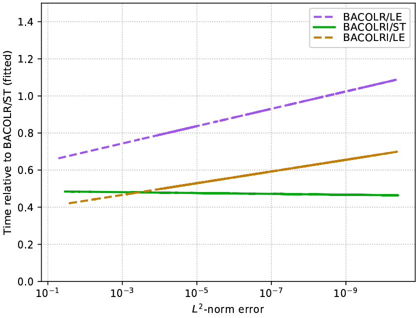


Figure: Error vs. CPU Time Relative to BACOLR/ST, TLBE $\epsilon = 10^{-4}$, $p = 6$

BACOLI vs. BACOLRI

- Coupled Nonlinear Schrödinger System
  - $(u_1)_t = i(\frac{1}{2}(u_1)_{xx} + \eta(u_1)_x + (|u_1|^2 + \rho|u_2|^2)u_1)),$
  - $(u_2)_t = i(\frac{1}{2}(u_2)_{xx} - \eta(u_2)_x + (\rho|u_1|^2 + |u_2|^2)u_2)),$
  - $-30 < x < 90, \quad t > 0$
  - Boundary conditions:
    $(u_1)_x(-30,t) = (u_2)_x(-30,t) = 0,$
    $(u_1)_x(90,t) = (u_2)_x(90,t) = 0, t > 0$
  - Initial conditions, $u_1(x,0)$, $u_2(x,0)$, chosen so that modulus for each exact solution component is a soliton
- $p = 6$, $tol = 10^{-2}, \ldots, 10^{-10}$
- BACOLI cannot solve this problem. Here BACOLI restricted to run with DASSL using only BDFs of orders 1 and 2

Software for 1D PDEs: B-spline Gaussian Collocation

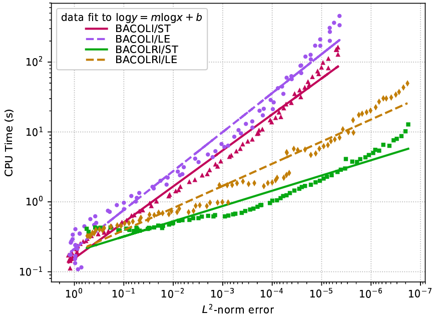


Figure: Error vs. CPU Time for BACOLR/ST, BACOLR/LE, BACOLRI/ST, and BACOLRI/LE, Coupled Nonlinear Schrödinger System, $p = 6$

Software for 1D PDEs: B-spline Gaussian Collocation

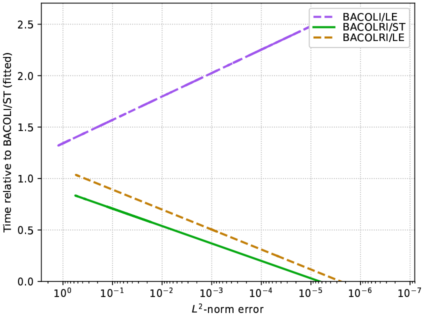


Figure: Error vs. CPU Time Relative to BACOLR/ST, Coupled Nonlinear Schrödinger System, $p = 6$

Summary + Future Work

I: 1D PDEs: Gaussian Collocation Error Control Software:

- **BACOLRI:** Latest member of adaptive error control Gaussian collocation software for 1D PDEs
- Improves on previous codes within this family:
  - Improves on spatial error estimation scheme of BACOLR: SCI and LOI
  - Improves on BACOLI: better stability through the use of RADAU5
- Additional modifications re LOI spatial error estimate, automatic selection of p and error control mode, etc.
- Error control B-spline Gaussian collocation algorithms for 1D PDEs $\Rightarrow$ On-going investigation of error control B-spline Gaussian collocation algorithms for 2D PDEs

Summary + Future Work

II: Gaussian Collocation Error Control Software for 2D PDEs:

- Tensor product B-spline Gaussian collocation on 2D rectangular grids [Li, Muir, 2013] (Already completed)
- Requires uniform rectangular grids
- Spatial error estimation for 2D:
  - Tensor product SCI scheme - 2D piecewise bivariate polynomial
  - Tensor product LOI scheme - 2D piecewise bivariate polynomial

Summary + Future Work

- Spatial adaptivity and spatial error control after every accepted time step:
  - Adaptivity via moving mesh framework (involving equidistribution of spatial error estimate)
  - Solution of MMPDE gives transformation function $\Rightarrow$ transform physical PDEs to computational domain Ω_C
  - Solve transformed PDEs on uniform rectangular grid in Ω_C
  - Change number of mesh points, N, M, to control spatial error estimates $<$ tolerance

- **Thank You**