Spectral Deferred Correction (SDC)

A. Donev, Fall 2021

These brief notes on SDC are based on an extensive body of work in the group of Mike Minion, based on initial work by Rokhlin & Greengard. Making SDC efficient & spectrally accurate for PDEs is hard & requires a host of other "tricks" that we won't have time to cover.

We want to solve the ODEs, $\varphi(0) = \varphi_0$ and

$$\varphi'(t) = F(t, \varphi(t))$$

to very high accuracy (instead of thinking about order, let's say we want 6-9 digits of accuracy)

These ODEs could come from a (pseudo) spectral discretization of a PDE that is very accurate, so we don't want our temporal integration error to completely dominate (as in your HW3).

Key idea is to convert ODE into an integral equation (this we will do for elliptic PDEs in boundary integral methods, and can also be useful for BVPs, see "Spectral integration & two-point BVPs" by Leslie Greengard, 1991):

$$\varphi(t) = \varphi_0 + \int_0^t F(\tau, \varphi(\tau))\, d\tau$$

$0 \le t \le \Delta t$ (Picard eq.)

Assume we solve the ODE using some standard predictor ODE solver to get a guess solution $\varphi^{(0)}(t)$

[While ODE solvers will only provide $\varphi^{(0)}$ at discrete points, we can always use (polynomial) interpolation to turn into a function]

The error
$$\delta(t) = \varphi(t) - \varphi^{(0)}(t)$$
satisfies the integral equation

$$\delta(t) = \int_0^t \Big[F\big(\tau, \varphi^{(0)}(\tau) + \delta(\tau)\big) - F\big(\tau, \varphi^{(0)}(\tau)\big) \Big] \, d\tau \qquad (*)$$
$$+ \underbrace{\Big[\varphi_0 - \varphi^{(0)}(t) + \int_0^t F\big(\tau, \varphi^{(0)}(\tau)\big) d\tau \Big]}_{\text{residual } \varepsilon(t)}$$

We can discretize the integrals using a spectral quadrature like Gauss quadrature. Then (*) is a (large) non-linear system of equations for the correction $\delta(t_j)$ where $\{t_j\}$ are the quadrature nodes. If we managed to solve (*) to sufficient accuracy we would have a spectrally-accurate temporal integrator / ODE solver!

How do we solve (*)?

Note: We didn't have to start with a predictor $\varphi^{(0)}$ per se, we could have just discretized the Picard equation:

$$\vec{\varphi} = \vec{\varphi}_0 + \Delta t \overleftrightarrow{S} \vec{F}(\vec{\varphi})$$

where $\overleftrightarrow{S}$ is the spectral integration matrix.

Define the residual

$$\varepsilon(\underset{\sim}{\vec{\varphi}}) = \left(\vec{\varphi}_0 + \Delta t\, S \vec{F}(\underset{\sim}{\vec{\varphi}})\right) - \underset{\sim}{\vec{\varphi}}$$

Here Δt is a "large" time step size chosen based on memory requirements & stability

current guess for $\varphi(\{t_k\})$

Recall (*):

$$\delta(t) = \int_0^t \left(F(\tau, \underset{\sim}{\varphi} + \delta) - F(\tau, \underset{\sim}{\varphi}) \right) d\tau$$
$$+ \mathcal{E}(t)$$

Now, this is itself an ODE written in integral form,

$$\delta(t_{m+1}) = \delta(t_m) + \left(\mathcal{E}(t_{m+1}) - \mathcal{E}(t_m) \right)$$
$$+ \int_{t_m}^{t_{m+1}} \left[F(\tau, \underset{\sim}{\varphi} + \delta) - F(\tau, \underset{\sim}{\varphi}) \right] d\tau$$

The analog of solving the correction ODE using forward / backward Euler is the FE / BE pass:

$$\delta(t_{m+1}) = \delta(t_m) + \left(\varepsilon(t_{m+1}) - \varepsilon(t_m)\right)$$

(**)

$$\left(t_{m+1} - t_m\right)\left[F\left(\tau, \tilde{\varphi} + \delta(t_{m/m+1})\right) - F(\tau, \tilde{\varphi})\right] d\tau$$

Forward Euler (m) Backward Euler ($m+1$)

with initial condidition

$$\delta(t_0) = \varepsilon(t_0) \; [= 0]$$

(if the IC of original ODE is satisfied exactly by $\tilde{\varphi}$)

This gives us a corrected solution $\varphi \simeq \tilde{\varphi} + \delta$. We can iterate this multiple times as a sort of fixed-point iteration:

SDC iteration:

1) Use some predictor method to solve ODE, for example, use Forward / Backward Euler depending on whether the ODE is non-stiff / stiff & get predicted solution

$$\varphi^{(0)} = \varphi(\{t_k\})$$

Gauss or other spectral nodes on $[0, \Delta t]$

Note: Whether the grid of quadrature nodes includes or not the left/right endpoint affects stability [for FE include left, but for BE include right]

2) Iterate until "convergence":

a) Compute $\mathcal{E}^{(k)} = \mathcal{E}(\vec{\varphi}^{(k)})$ using spectral quadrature

b) Compute correction $\vec{\delta}^{(k)}$ with FE/BE using $(\ast\ast)$

(this is like solving ODE using FE/BE on a non-uniform grid of time points)

c) Correct solution

$$\varphi^{(k+1)} = \varphi^{(k)} + \delta^{(k)}$$

This iteration is found to, with suitable choice of quadrature nodes, inherit the stability of FE/BE.

Properties for linear problems for sufficiently small Δt:

a) the SDC iteration converges

b) Each iteration increases the order of accuracy (in Δt) by 1, up to the maximal order of the quadrature rule.

So SDC is a way to get any order of accuracy we desire, without having to derive complicated RK formulas, for example. But it may be expensive (especially in memory for PDEs)

A useful way to think of SDC iteration is as a preconditioned iterative method to solve

$$\vec{\varphi} = \vec{\varphi_0} + \Delta t \overleftrightarrow{S} \vec{F}(\vec{\varphi})$$

where FE/BE is the preconditioner. The simple SDC iteration is a fixed-point type method but for linear problems one can easily use GMRES, for example, at the expense of extra memory [see Huang, Jia, Minion, 2006]