1 Collocation

Consider an IVP in Picard formulation over a single timestep $[T_n, T_{n+1}]$,

$$\tilde{u}(t) = \tilde{u}(T_n) + \int_{T_n}^{t} f(\tilde{u}(s), s)\, ds,\ T_n \le t \le T_{n+1}. \tag{1}$$

Now let $(t_m)_{m=1,\ldots,M}$ denote a set of M collocation nodes in the interval $[T_n, T_{n+1}]$ with

$$T_n \le t_1 < \ldots, < t_M \le T_{n+1}. \tag{2}$$

Also, define $t_0 := T_n$.

Remark 1. *For Gauss-Legendre nodes, it is $T_n = t_0 < t_1$ and $t_M < T_{n+1}$. For Gauss-Radau nodes it is $T_n = t_0 < t_1$ but $t_M = T_{n+1}$. For Gauss-Lobatto nodes it is $T_n = t_0 = t_1$ and $t_M = T_{n+1}$.*

Denote as u the polynomial of degree M satisfying the collocation conditions

$$u'(t_m) = f(u(t_m), t_m),\ m = 1, \ldots, M. \tag{3}$$

and the initial condition

$$u(T_n) = U_0 \tag{4}$$

with $U_0 \approx \tilde{u}(T_n)$ coming from the previous timestep. The polynomial u has the representation

$$u(t) = \sum_{m=1}^{M} U_m l_j(t) \tag{5}$$

where $U_m = u(t_m)$ are unknown coefficients and $(l_j)_{j=1,\ldots,M}$ are the Lagrange polynomials defined by the collocation nodes $(t_m)_{m=1,\ldots,M}$. Replace $\tilde{u}$ in (1) with u for $t = t_m$, $m = 1, \ldots, M$ to get the equations

$$u(t_m) = U_m = u(T_n) + \int_{T_n}^{t_m} u'(s)\, ds \tag{6}$$

$$= U_0 + \Delta t \sum_{j=1}^{M} q_{m,j} u'(t_m) \tag{7}$$

$$= U_0 + \Delta t \sum_{j=1}^{M} q_{m,j} f(u(t_m), t_m) \tag{8}$$

so that using $u(t_m) = U_m$ we get

$$U_m = U_0 + \Delta t \sum_{j=1}^{M} q_{m,j} f(U_j, t_j) \tag{9}$$

with

$$q_{m,j} := \frac{1}{\Delta t} \int_{T_n}^{t_m} l_j(s)\, ds \tag{10}$$

being quadrature weights and $\Delta t := T_{n+1} - T_n$. Denote furthermore the approximation at the endpoint T_{n+1} of the timestep as

$$U_{M+1} := u(T_{n+1}) \approx \tilde{u}(T_{n+1}). \tag{11}$$

Once the coefficients $(U_m)_{m=1,\ldots,M}$ are known, U_{M+1} can be computed from

$$U_{M+1} = U_0 + \Delta t \sum_{j=1}^{M} q_{M+1,j} f(U_j, t_j) \tag{12}$$

with

$$q_{M+1,j} := \frac{1}{\Delta t} \int_{T_n}^{T_{n+1}} l_j(s)\, ds. \tag{13}$$

Remark 2. *For Gauss-Radau nodes with $T_{N+1} = t_M$, it follows from* (10) *and* (13) *that $q_{M,j} = q_{M+1,j}$ for $j = 1, \ldots, M$. Therefor, from* (9) *and* (12)*, it also follows that $U_M = U_{M+1}$.*

Remark 3. *For Gauss-Lobatto nodes with $T_N = t_0 = t_1$ and $t_M = T_{n+1}$, it is $U_M = U_{M+1}$ as explained in Remark 2. Additionally, from* (10)*, it also follows that $q_{1,j} = 0$ for $j =, 1, \ldots, M$ and therefor, from* (9)*, that $U_1 = U_0$.*

Collecting the unknowns $(U_m)_{m=1,\ldots,M}$ in a vector

$$\mathbf{U} := [U_1, \ldots, U_M]^T, \tag{14}$$

their function values in a vector

$$\mathbf{F}(\mathbf{U}) := [f(U_1, t_1), \ldots, f(U_M, t_M)]^T \tag{15}$$

and the quadrature weights $q_{m,j}$ in an $M \times M$ matrix

$$\mathbf{Q} = (q_{m,j})_{m,j=1,\ldots,M} \tag{16}$$

allows to compactly write the M equations (9) as

$$\mathbf{U} = \mathbf{U}_0 + \Delta t \mathbf{Q} \mathbf{F}(\mathbf{U}) \tag{17}$$

with $\mathbf{U}_0 = [U_0, \ldots, U_0]^T \in \mathbb{R}^M$.

Remark 4. *One can include equation* (12) *for the end value U_{M+1} into* (17) *by defining*

$$\hat{\mathbf{U}} = [U_1, \ldots, U_M, U_{M+1}]^T \in \mathbb{R}^{M+1}, \tag{18}$$

the vector of function values

$$\hat{\mathbf{F}}(\hat{\mathbf{U}}) := [f(U_1, t_1), \ldots, f(U_M, t_M), 0]^T \tag{19}$$

the $M+1 \times M+1$ matrix

$$\hat{\mathbf{Q}} := \begin{pmatrix} q_{1,1} & \ldots & q_{1,M} & 0 \\ \vdots & & \vdots & \vdots \\ q_{M+1,1} & \ldots & q_{M+1,M} & 0 \end{pmatrix} \tag{20}$$

leading to

$$\hat{\mathbf{U}} = \hat{\mathbf{U}}_0 + \Delta t \hat{\mathbf{Q}} \hat{\mathbf{F}}(\hat{\mathbf{U}}) \tag{21}$$

with $\hat{\mathbf{U}}_0 := [U_0, \ldots, U_0]^T \in \mathbb{R}^{M+1}$. Note that because the last column of $\hat{\mathbf{Q}}$ contains only zeros, the value in the last entry for $\hat{\mathbf{F}}(\hat{\mathbf{U}})$ is arbitrary.

Remark 5. *The value of the interpolation polynomial u can be reconstructed at any value $T_n \leq t \leq T_{n+1}$ from the formula*

$$u(t) = U_0 + \sum_{m=1}^{M} \tilde{q}_m U_m \tag{22}$$

with quadrature weights

$$\tilde{q}_m := \frac{1}{\Delta t} \int_{T_n}^{t} l_m(s)\, ds. \tag{23}$$

Remark 6. *At least in theory, values $f(U_j, t_j)$ could be reconstructed from*

$$f(U_j, t_j) = u'(t_j) = \sum_{m=1}^{M} U_m l'_m(t_j) \tag{24}$$

if the values $l'_m(t_j)$ are known. **However:** *Note that this is true only for the actual collocation solution, not the iterative approximations provided by SDC. So its more kind of an approximate f evaluation.*

2 Picard iteration

An easy and simple approach to solve the system (17) would be Picard iteration

$$\mathbf{U}^{k+1} = \mathbf{U}_0 + \Delta t \mathbf{Q}\mathbf{F}(\mathbf{U}^k) \tag{25}$$

with initial guess $\mathbf{U}^0 = \mathbf{U}_0$ for example. Component wise, the above iteration reads

$$U_m^{k+1} = U_0 + \Delta t \sum_{j=1}^{M} q_{m,j} f(U_j^k, t_j),\ m = 1, \dots, M. \tag{26}$$

Define now weights of quadrature rules that approximate node-to-node integrals, that is

$$s_{m,j} := \frac{1}{\Delta t} \int_{t_{m-1}}^{t_m} l_j(s)\, ds \tag{27}$$

for $j = 1, \dots, M$ and $m = 1, \dots, M$, keeping in mind the convention that $t_0 = T_n$.

Remark 7. *By definition* (10) *and* (27)*, it is* $q_{1,j} = s_{1,j}$ *for* $j = 1, \dots, M$.

Proposition 1. *For any* $1 \leq j \leq M$ *and* $1 \leq m \leq M+1$ *it is*

$$q_{m,j} = \sum_{i=1}^{m} s_{i,j}. \tag{28}$$

Proof. Follows immediately from definitions (10) and (27). □

Proposition 2. *For* $\mathbf{Q} \in \mathbb{R}^{M,M}$ *with entries* (10) *and* $\mathbf{S} \in \mathbb{R}^{M,M}$ *with entries* (27) *it is*

$$\mathbf{Q} = \begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \dots & 1 \end{pmatrix} \mathbf{S} \tag{29}$$

Proof. Follows from proposition 1.[TODO – check...] □

Proposition 3. *For any* $1 \leq j \leq M$ *and* $2 \leq m \leq M+1$ *it is*

$$q_{m,j} - q_{m-1,j} = s_{m,j} \tag{30}$$

Proof. Follows immediately from Proposition 1 □

Now, we can reformulate the component wise Picard iteration (26) using the node-to-node integration formulas.

Lemma 1. *The component wise Picard iteration* (26) *is equivalent to the node-to-node sweep*

$$U_m^{k+1} = U_{m-1}^{k+1} + \Delta t \sum_{j=1}^{M} s_{m,j} f(U_j^k, t_j),\ m = 1 \dots, M \tag{31}$$

with $U_0^{k+1} = U_0$ *for all* $k \geq 0$.

Proof. For $2 \leq m \leq M$, subtract (26) for m and $m-1$ to get

$$U_m^{k+1} - U_{m-1}^{k+1} = \Delta t \sum_{j=1}^{M} q_{m,j} f(U_j^k, t_j) - \Delta t \sum_{j=1}^{M} q_{m-1,j} f(U_j^k, t_j) \tag{32}$$

$$= \Delta t \sum_{j=1}^{M} (q_{m,j} - q_{m-1,j}) f(U_j^k, t_j) \tag{33}$$

$$= \Delta t \sum_{j=1}^{M} s_{m,j} f(U_j^k, t_j) \tag{34}$$

using Proposition 3. For $m = 1$, the proposition follows from $U_0^{k+1} = U_0$, the fact that $q_{1,j} = s_{1,j}$ and (26) with $m = 1$. □

3 SDC sweeps

It is possible to accelerate convergence of the Picard iteration by preconditioning. Introduce the $M \times M$ matrix

$$\mathbf{Q}_\Delta := \frac{1}{\Delta t} \begin{pmatrix} \Delta t_1 & 0 & & \\ \Delta t_1 & \Delta t_2 & 0 & \\ \vdots & \vdots & & \\ \Delta t_1 & \Delta t_2 & \dots & \Delta t_M \end{pmatrix} \tag{35}$$

Proposition 4. *The matrix $\mathbf{Q}_\Delta$ can be written as*

$$\mathbf{Q}_\Delta = \begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} \Delta t_1 & & & \\ & \Delta t_2 & & \\ & & \ddots & \\ & & & \Delta t_M \end{pmatrix} \tag{36}$$

Multiplication by $\mathbf{Q}_{\Delta t}$ corresponds to a composite right hand side quadrature rule, with row m approximating the integral over $[T_n, t_m]$

$$\Delta t \left(\mathbf{Q}_\Delta \mathbf{U}\right)_m = \sum_{j=1}^{m} \Delta t_j U_j \approx \int_{T_0}^{t_m} u(s)\, ds. \tag{37}$$

Note that because the right hand side rule is only of order one, this is really just an approximation of the integral, even though u is a polynomial. The analogue to (17) with some given right hand side vector $\mathbf{B}$ reads

$$\mathbf{U} = \mathbf{U}_0 + \Delta t \mathbf{Q}_\Delta \mathbf{F}(\mathbf{U}) + \mathbf{B} \tag{38}$$

that is

$$\mathbf{U} - \Delta t \mathbf{Q}_\Delta \mathbf{F}(\mathbf{U}) = \mathbf{U}_0 + \mathbf{B} \tag{39}$$

or, component wise,

$$U_m = U_0 + \sum_{j=1}^{m} \Delta t_j f(U_j, t_j) + B_m,\ m = 1, \dots, M. \tag{40}$$

The following Lemma shows that (38), in contrast to (25), is easy to solve.

Lemma 2. *The component wise formula* (40) *is equivalent to an implicit Euler sweep*

$$U_m = U_{m-1} + \Delta t_m f(U_m, t_m) + B_m - B_{m-1},\ m = 1, \dots, M. \tag{41}$$

with $B_0 := 0$.

Proof. For $2 \leq m \leq M$, take the difference of (40) for m and $m-1$ to get

$$U_m - U_{m-1} = \Delta t_m f(U_m, t_m) + B_m - B_{m-1}. \tag{42}$$

For $m = 1$, (40) directly shows the proposition. □

We can use $\mathbf{Q}_\Delta$ to precondition the Picard iteration, which leads to the following iteration

$$\mathbf{U}^{k+1} - \Delta t \mathbf{Q}_\Delta \mathbf{F}(\mathbf{U}^{k+1}) = \mathbf{U}_0 + \Delta t \left(\mathbf{Q} - \mathbf{Q}_\Delta\right) \mathbf{F}(\mathbf{U}^k). \tag{43}$$

The component wise formulation of this iteration corresponds to an SDC sweep, as is shown by the following.

Lemma 3. *The component wise iteration* (43) *is equivalent to an SDC sweep with an implicit Euler, that is*

$$U_m^{k+1} = U_{m-1}^{k+1} + \Delta t_m \left(f(U_m^{k+1}, t_m) - f(U_m^k, t_m) \right) + \Delta t \sum_{j=1}^{M} s_{m,j} f(U_j^k, t_j). \tag{44}$$

Proof. First, define $\mathbf{B} := \Delta t \left(\mathbf{Q} - \mathbf{Q}_\Delta\right) \mathbf{U}^k$ as right hand side. Then, component m reads

$$B_m = \Delta t \sum_{j=1}^{M} q_{m,j} f(U_j^k, t_j) - \sum_{j=1}^{m} \Delta t_j f(U_j^k, t_j) \tag{45}$$

so that for $2 \leq m \leq M$, using Proposition 3, it is

$$B_m - B_{m-1} = \Delta t \sum_{j=1}^{M} s_{m,j} f(U_j^k, t_j) - \Delta t_m f(U_m^k, t_m) \tag{46}$$

Further, for $m = 1$ by Remark 7 it is $q_{1,j} = s_{1,j}$ and therefor with $B_0 := 0$ we get

$$B_1 - B_0 = \Delta t \sum_{j=1}^{M} s_{1,j} f(U_j^k, t_j) - \Delta t_1 f(U_1^k, t_1). \tag{47}$$

Then, the solution of (43) according to Lemma 2 reads

$$\begin{aligned} U_m^{k+1} &= U_{m-1}^{k+1} + \Delta t_m f(U_m^{k+1}, t_m) + B_m - B_{m-1} && (48) \\ &= U_{m-1}^{k+1} + \Delta t_m \left(f(U_m^{k+1}, t_m) - f(U_m^k, t_m) \right) + \Delta t \sum_{j=1}^{M} s_{m,j} f(U_j^k, t_j) && (49) \end{aligned}$$

for $1 \leq m \leq M$ which completes the proof. □

3.1 SDC(θ) methods

If the matrix $\mathbf{Q}_\Delta$ is weighted by a coefficient θ, the resulting iteration becomes

$$\mathbf{U}^{k+1} - \Delta t \theta \mathbf{Q}_\Delta \mathbf{F}(\mathbf{U}^{k+1}) = \mathbf{U}_0 + \Delta t \left(\mathbf{Q} - \theta \mathbf{Q}_\Delta\right) \mathbf{F}(\mathbf{U}^k) \tag{50}$$

or, component wise,

$$U_m^{k+1} = U_{m-1}^{k+1} + \theta \Delta t_m \left(f(U_m^{k+1}, t_m) - f(U_m^k, t_m) \right) + \Delta t \sum_{j=1}^{M} s_{m,j} f(U_j^k, t_j) \tag{51}$$

[TODO – check this...] For $\theta = 1$, the original SDC method is retrieved while for $\theta = 0$ the iteration reduces to Picard.