1 The matrices S and Q in SDC

Assume we have a given time-interval $[T_0, T_1]$ and a set of collocation nodes τ_m, $m = 0, \ldots, M$ with

$$T_0 \leq \tau_0 < \ldots < \tau_M \leq T_1 \tag{1}$$

For Gauss-Lobatto quadrature, τ_0 and τ_M will coincide with T_0 and T_1, for Gauss-Legendre quadrature they won't.

1.1 Quadrature over the full interval

For a set of given nodes $(\tau_j)_{j=0,\ldots,M}$, there is a unique set of weights q_m, $m = 0, \ldots, M$ such that the corresponding quadrature rule yields an approximation of the integral over $[T_0, T_1]$ with maximum order $2M - 2$ for Gauss-Lobatto, $2M$ for Gauss-Legendre and $2M - 1$ for Gauss-Radau. That is with

$$\mathbf{q} := (q_0, \ldots, q_M) \in \mathbb{R}^{M+1} \tag{2}$$

we have

$$\int_{T_0}^{T_1} u(\tau)\, d\tau \approx \sum_{m=0}^{M} q_m u_m = \mathbf{q}^{\mathrm{T}} \cdot \mathbf{u} =: \mathbf{I}(\mathbf{u}) \tag{3}$$

for

$$\mathbf{u} := (u_0 \ldots, u_M) \in \mathbb{R}^{M+1} \text{ with } u_j \approx u(\tau_j),\ j = 0, \ldots, M. \tag{4}$$

Denoting by l_m, $m = 0, \ldots M$, the Lagrangian polynomials with roots τ_m, the weights q_m are given as

$$q_m = \int_{T_0}^{T_1} l_m(\tau)\, d\tau \tag{5}$$

1.2 Quadrature from left endpoint to a collocation node – Q

The Picard formula on which SDC is based reads for a point t in $[T_0, T_1]$

$$u(t) = u(T_0) + \int_{T_0}^{t} f(u(\tau), \tau)\, d\tau. \tag{6}$$

Thus, the solution at some collocation node τ_m is given by

$$u(\tau_m) = u(T_0) + \int_{T_0}^{\tau_m} f(u(\tau), \tau)\, d\tau. \tag{7}$$

To obtain a discrete version, we now define operators $\mathbf{I}_0^m$ that provide an approximation of the integral over a part of the time interval $[T_0, T_1]$, namely

$$\mathbf{I}^m(\mathbf{u}) := \mathbf{q}_m^{\mathrm{T}} \cdot \mathbf{u} = \sum_{j=0}^{M} q_{m,j} u(\tau_j) \approx \int_{T_0}^{\tau_m} u(\tau)\, d\tau, \quad m = 0, \ldots, M. \tag{8}$$

Remark 1. *If the left endpoint T_0 coincides with the first collocation node τ_0, we get $\mathbf{I}_0^0 \equiv 0$.*

Remark 2. *If the right endpoint T_1 coincides with the last collocation node τ_M, we have $\mathbf{I} = \mathbf{I}_0^M$, cf. (**??**), but otherwise the two operators are different.*

The weights for the $\mathbf{I}_0^m$ can be computed from the following consideration: Let p denote the uniquely determined interpolation polynomial through the points (τ_m, u_m), so that

$$p(\tau) = \sum_{m=0}^{M} u_m l_m(\tau) \tag{9}$$

where l_m are the Lagrangian polynomials as introduced above. Then, approximating

$$\int_{T_0}^{\tau_m} u(\tau) d\tau \approx \int_{T_0}^{\tau_m} p(\tau)\, d\tau \tag{10}$$

leads to weights

$$q_{m,j} := \int_{T_0}^{\tau_m} l_j(\tau)\, d\tau. \tag{11}$$

Remark 3. *Because the l_j are polynomials of order M , the weights $q_{m,j}$ can be exactly computed using a quadrature rule of order $M + 1$.*

The matrix $\mathbf{Q}$ is now defined as

$$\mathbf{Q} := \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_M \end{bmatrix} \in \mathbb{R}^{M+1,M+1} \tag{12}$$

so that

$$\mathbf{Q}\mathbf{u} = \begin{pmatrix} \mathbf{I}^0(\mathbf{u}) \\ \mathbf{I}^1(\mathbf{u}) \\ \vdots \\ \mathbf{I}^M(\mathbf{u}) \end{pmatrix} \in \mathbb{R}^{M+1} \tag{13}$$

Remark 4. *The iteration*

$$\mathbf{u}^{k+1} = u(T_0)\mathbf{1} + \mathbf{Q} f(\mathbf{u}^k) \tag{14}$$

corresponds to a component wise iteration

$$u_m^k = u(T_0) + \mathbf{I}^m(f(\mathbf{u}^k)) \approx u(T_0) + \int_{T_0}^{\tau_j} f(u^k(\tau), \tau)\, d\tau, \quad m = 0, \ldots, M \tag{15}$$

and thus to a discrete version of the Picard iteration.

Remark 5. *For $\tau_M < T_1$, an approximation for $u(T_1)$ can easily be obtained in every iteration by computing*

$$u = u(T_0) + \mathbf{I}(f(\mathbf{u}^k)) = u(T_0) + \mathbf{q}^{\mathrm{T}} \cdot f(\mathbf{u}^k) \approx u(T_0) + \int_{T_0}^{T_1} f(u(\tau), \tau)\, d\tau. \tag{16}$$

Remark 6. *If $T_0 = \tau_0$, it is $\mathbf{I}^0 \equiv 0$ and thus the first row in $\mathbf{Q}$ consists of zeros. Iteration (**??**) thus never changes the first entry in $\mathbf{u}$, leading to*

$$u_0^{k+1} = u(T_0) \tag{17}$$

for all $k \geq 0$.

1.3 Quadrature between two consecutive collocation nodes – S

For use within the sweeps of SDC, we also need approximations to the integral between two consecutive nodes, i.e. an operator

$$\mathbf{I}_{m-1}^{m}(\mathbf{u}) =: \mathbf{s}_m \cdot \mathbf{u} = \sum_{j=0}^{M} s_{m,j} u(\tau_j) \approx \int_{\tau_{m-1}}^{\tau_m} u(\tau)\, d\tau, \quad m = 1, \ldots, M \tag{18}$$

As before, the weights can be computed as integrals over the Lagrangian polynomials l_j, that is

$$s_{m,j} = \int_{\tau_{m-1}}^{\tau_m} l_j(\tau)\, d\tau \tag{19}$$

Note that from (**??**) and (**??**) we immediately obtain the following identity

$$q_{m,j} = q_{0,j} + \sum_{j=1}^{m} s_{m,j} \Leftrightarrow \int_{T_0}^{\tau_m} l_j(\tau)\, d\tau = \int_{T_0}^{\tau_0} l_j(\tau)\, d\tau + \sum_{k=1}^{m} \int_{\tau_{k-1}}^{\tau_k} l_j(\tau)\, d\tau \tag{20}$$

and thus

$$\mathbf{q}_m = \mathbf{q}_0 + \sum_{j=1}^{m} \mathbf{s}_j \tag{21}$$

For $\mathbf{q}_m$, $m = 0, \ldots, M$ given, we can thus compute the weights $\mathbf{s}_m$, $m = 1, \ldots, M$ from

$$\mathbf{s}_0 := \mathbf{q}_0 \tag{22}$$
$$\mathbf{s}_1 = \mathbf{q}_1 - \mathbf{q}_0 \tag{23}$$
$$\mathbf{s}_2 = \mathbf{q}_2 - \mathbf{q}_0 - \mathbf{s}_1 = \mathbf{q}_2 - \mathbf{q}_1 \tag{24}$$
$$\mathbf{s}_3 = \mathbf{q}_3 - \mathbf{q}_0 - \mathbf{s}_1 - \mathbf{s}_2 = \mathbf{q}_3 - \mathbf{q}_2 \tag{25}$$
$$\vdots \tag{26}$$

where the definition of $\mathbf{s}_0 := \mathbf{q}_1$ is for convenience. This allow to immediately state the following identity

Lemma 1. *For any $1 \leq m \leq M$ it holds that*

$$\mathbf{I}^0 + \sum_{j=0}^{m-1} \mathbf{I}_j^{j+1}(\mathbf{u}) = \mathbf{I}^m(\mathbf{u}) \tag{27}$$

Proof. Use the identity $q_j = s_j + q_{j-1}$ shown above and compute

$$\mathbf{I}^m(\mathbf{u}) = \mathbf{q}_m^{\mathrm{T}} \cdot \mathbf{u} \tag{28}$$
$$= (\mathbf{s}_m + \mathbf{q}_{m-1})^{\mathrm{T}} \cdot \mathbf{u} \tag{29}$$
$$= (\mathbf{s}_m + \mathbf{s}_{m-1} + \mathbf{q}_{m-2})^{\mathrm{T}} \cdot \mathbf{u} \tag{30}$$
$$= \ldots \tag{31}$$
$$= (\mathbf{s}_m + \ldots + \mathbf{s}_1 + \mathbf{s}_0)^{\mathrm{T}} \cdot \mathbf{u} \tag{32}$$
$$= \mathbf{s}_m^{\mathrm{T}} \cdot \mathbf{u} + \ldots + \mathbf{s}_1^{\mathrm{T}} \cdot \mathbf{u} + \mathbf{s}_0^{\mathrm{T}} \cdot \mathbf{u} \tag{33}$$
$$= \mathbf{I}_{m-1}^{m}(\mathbf{u}) + \ldots + \mathbf{I}_0^1(\mathbf{u}) + \mathbf{I}^0(\mathbf{u}). \tag{34}$$

□

The matrix $\mathbf{S}$ is now defined as

$$\mathbf{S} := \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_M \end{bmatrix} \in \mathbb{R}^{M+1,M+1} \tag{35}$$

so that

$$\mathbf{Su} = \begin{pmatrix} \mathbf{I}^0 \\ \mathbf{I}_0^1 \\ \mathbf{I}_1^2 \\ \vdots \\ \mathbf{I}_{M-1}^M \end{pmatrix} \in \mathbb{R}^{M+1} \tag{36}$$

when defining

$$\mathbf{I}_{j-1}^j(\mathbf{u}) := \mathbf{s}_j^{\mathrm{T}} \cdot \mathbf{u},\ j = 1, \ldots, M \text{ and } \mathbf{I}^0(\mathbf{u}) := \mathbf{s}_0^{\mathrm{T}} \cdot \mathbf{u}. \tag{37}$$

2 Picard iteration

As shown in Remark **??**, the matrix $\mathbf{Q}$ can be used to define a discrete Picard iteration. The iteration can be equivalently formulated as a "sweep" over the nodes.

Lemma 2. *The node-by-node sweep*

$$u_m^{k+1} = u_{m-1}^{k+1} + \mathbf{I}_{m-1}^m(f(\mathbf{u}^k)),\ m = 1, \ldots, M \tag{38}$$

with initialization

$$u_0^{k+1} := u(T_0) + \mathbf{I}^0(f(\mathbf{u}^k)) \tag{39}$$

is equivalent to the full-system iteration (**??**)*.*

Proof. For a $0 \leq m \leq M$, compute

$$\begin{aligned} u_m^{k+1} &= u_{m-1}^{k+1} + \mathbf{I}_{m-1}^m(f(\mathbf{u}^k)) && (40)\\ &= u_{m-2}^{k+1} + \mathbf{I}_{m-2}^{m-1}(f(\mathbf{u}^k)) + \mathbf{I}_{m-1}^m(f(\mathbf{u}^k)) && (41)\\ &= \ldots && (42)\\ &= u_0^{k+1} + \sum_{j=0}^{m-1} \mathbf{I}_j^{j+1}(f(\mathbf{u}^k)) && (43)\\ &= u(T_0) + \mathbf{I}^0(f(\mathbf{u}^k)) + \sum_{j=0}^{m-1} \mathbf{I}_j^{j+1}(f(\mathbf{u}^k)) && (44)\\ &= u(T_0) + \mathbf{I}^m(f(\mathbf{u}^k)) && (45) \end{aligned}$$

where the last identity used Lemma **??**. This is now exactly the component-wise Picard iteration (**??**). □

Remark 7. *Note that if the first node coincides with the left boundary, i.e.* $\tau_0 = T_0$*, then* $\mathbf{I}^0 \equiv 0$ *and the initialization* (**??**) *simply sets* $u_0^{k+1} = u(T_0)$*.*

Remark 8. *The product* $\mathbf{I}_{m-1}^m(f(\mathbf{u}^k))$ *in* (**??**) *corresponds to the scalar product*

$$\mathbf{s}_m^{\mathrm{T}} \cdot f(\mathbf{u}^k) = \sum_{j=0}^{M} s_j f(u_j^k) \tag{46}$$

In the object oriented implementation, this scalar product can be evaluated using the sum and scalar multiplication routine of the solution object, provided that if u *is a solution object, so is* $f(u)$*.*

3 SDC sweeps