

*Supporting Information for:*

# **Flow and Forced Convection Heat and Mass Transfer Characteristics of Developed Laminar Flow in Square Channels with Rounded Corners: A Model for Flow in Washcoated Monolith Channels<sup>1</sup>**

Timothy C. Watling

*Johnson Matthey, Blount's Court, Sonning Common, Reading RG4 9NH, UK*

[tim.watling@matthey.com](mailto:tim.watling@matthey.com)

## **Contents**

- S1. Geometric Calculations
- S2. Methodology
- S3. Convergence Behaviour
- S4. The Velocity and Temperature Field
- S5. Tabulated Data Characterising Fully Developed Laminar Flow
- S6. MATLAB<sup>®</sup> Program

---

<sup>1</sup> Published in Industrial & Engineering Chemistry Research.

## Nomenclature

Matrix variables are indicated in bold.

<b>A</b>	Column matrix whose elements are $a_j$ ; defined by Eq. (S23)
$A_C$	Cross-sectional area of the coated channel, $m^2$
$a_j$	Constant in Eq. (S20), $m^{2-jN}$
<b>C</b>	Column matrix whose elements are $c_j$ ; defined by Eq. (S49)
$c_j$	Constant in Eq. (S46), $m^{4-jN}$
$C_p$	Specific heat capacity of gas at constant pressure, $J\ kg^{-1}\ K^{-1}$
<b>D</b>	Matrix defined by Eq. (S24)
$d$	Width of the coated channel, <i>i.e.</i> distance between opposing straight sides of the channel, m
$d_H$	Hydraulic diameter of coated channel, $4A_C/p$ , m
$d_{UC}$	Width of the uncoated channel, m
<b>E</b>	Matrix defined by Eq. (S50)
$F$	Viscous loss coefficient, -
$f$	(Fanning) friction factor, -
$g(z)$	The $z$ dependant part of the temperature solution, K
$h$	Heat transfer coefficient for heat transfer between gas in the channel and the channel wall, $W\ m^{-2}\ K^{-1}$
$i$	An integer; positive, negative or zero, -
$j, k, l$	Indices in summations, non-negative integers, -
$L_{WC}$	Washcoat loading, $kg\ m^{-3}$
$m$	Number of points on the channel wall at which the boundary condition is evaluated, -
$N$	Order of rotational symmetry, -
$n$	Upper limit on $j, k$ and $l$ in the summations, -
$Nu$	Nusselt number, $hd_H/\lambda$ , -
$P$	Pressure of gas in the channel, Pa
$p$	Perimeter of the coated channel, m
$r$	Radial coordinate in polar coordinates relative to the channel centre, m
$r_C$	Radius of curvature of the channel corners, m
$Re$	Reynolds number, $\rho V_m d_H/\mu$ , -
$r_T$	Radial coordinate of the channel wall, m
<b>S</b>	Matrix defined by Eq. (S25)
$T$	Local temperature of the gas in the channel, K

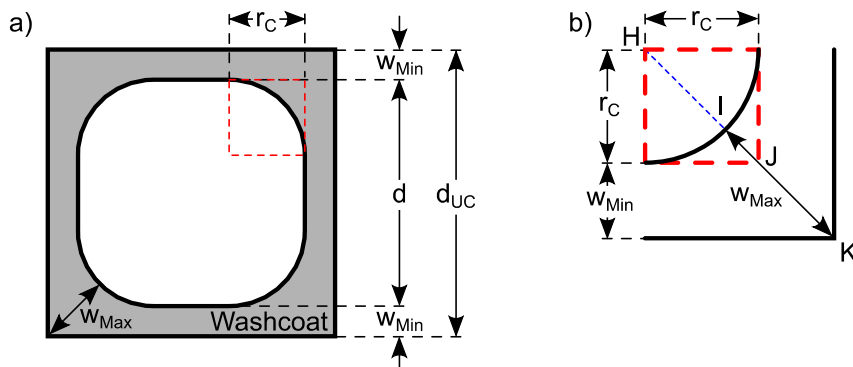
$T^*$	Normalised local temperature, defined by Eq. (S43), m <sup>4</sup>
$T_m$	Mixing cup temperature of gas in the channel, K
$T_m^*$	Mixing cup version of temperature $T^*$ , m <sup>4</sup>
$T_\Gamma$	Temperature of the channel wall, K
$T_{\Gamma,max}^*$	The maximum deviation of the normalised temperature at the channel wall from the intended boundary value of zero, m <sup>4</sup>
$\mathbf{U}$	Matrix defined by Eq. (S51)
$V$	Local axial gas velocity, m s <sup>-1</sup>
$V_m$	Mean axial velocity of gas in the channel, $\langle V \rangle$ , m s <sup>-1</sup>
$V_{max}$	Maximum gas velocity, m s <sup>-1</sup>
$V_{\Gamma,max}$	The maximum deviation of the gas velocity at the channel wall from the intended boundary value of zero, m s <sup>-1</sup>
$w_{Max}$	Maximum thickness of the washcoat, m
$w_{Min}$	Minimum thickness of the washcoat, m
$w_W$	Wall thickness of the monolith, m
$z$	Axial distance or coordinate, m
$\alpha$	Momentum flux correction factor, $\langle V^2 \rangle / \langle V \rangle^2$ , -
$\alpha_E$	Kinetic energy flux correction factor, $\langle V^3 \rangle / \langle V \rangle^3$ , -
$\gamma$	Channel shape factor, $2r_c/d$ , -
$\epsilon_C$	Open frontal area of the coated monolith, <i>i.e.</i> the open cross-sectional area of the face of the coated monolith divided by the total face area, -
$\epsilon_{UC}$	Open frontal area of the uncoated monolith, <i>i.e.</i> the open cross-sectional area of the face of the uncoated monolith divided by the total face area, -
$\theta$	Angular coordinate in polar coordinates, rad
$\lambda$	Thermal conductivity of the gas in the channel, W m <sup>-1</sup> K <sup>-1</sup>
$\mu$	Gas viscosity, Pa s
$\rho$	Gas density, kg m <sup>-3</sup>
$\rho_{Cell}$	Cell density, <i>i.e.</i> number of channels per unit cross-sectional area of the monolith, m <sup>-2</sup>
$\rho_{WC}$	Density of the washcoat after coating, kg m <sup>-3</sup>
$\Psi(r, \theta)$	The $r$ and $\theta$ dependant part of the temperature solution, K
$\langle \ \rangle$	Average of the bracketed quantity over a cross section

## S1. Geometric Calculations

The channel shape factor,  $\gamma$ , is defined in the main paper (Eq. (4)) as:

$$\gamma = \frac{2r_c}{d} \quad (\text{S1})$$

where  $r_c$  is the radius of curvature of the round corners and  $d$  is the width of the coated channel as shown in Fig. S1.



**Fig. S1** Geometry of a coated monolith channel, assuming the coated channel is a square with rounded corners. a) Cross-section of the whole channel: the outer square represents the uncoated channel and the grey area the washcoat. b) Close up of a corner of the channel used for calculation of  $w_{Max}$ . This is a schematic; the washcoat thickness is not intended to be representative.

### S1.1. Channel Perimeter and Cross-Sectional Area as a Function of $\gamma$

To calculate the perimeter and cross-sectional area of a square channel with rounded corners, consider the channel as being formed from a square channel by removing a square of side  $r_c$  from each of the four corners and replacing it with a quadrant of radius  $r_c$  (Fig. 1a in the main paper, Fig. S1). Thus, the cross-sectional area,  $A_C$ , is obtained by taking a square of area  $d^2$  and for each of the four corners subtracting the area of the square removed ( $r_c^2$ ) and adding the area of the quadrant ( $\pi r_c^2/4$ ). This gives:

$$A_C = d^2 - 4r_c^2 + 4(\pi r_c^2/4) \quad (\text{S2})$$

Note the “4”s in this equation which arise from there being four corners. Substitution of Eq. (S1) gives:

$$A_C = (1 - [1 - \pi/4]\gamma^2)d^2 \quad (\text{S3})$$

which is Eq. (6) in the main paper.

Similarly, the channel perimeter,  $p$ , is obtained by taking the square (perimeter  $4d$ ) and for each of the four corners subtracting the external perimeter of the square removed ( $2r_c$ ) and adding the curved perimeter of the quadrant ( $\pi r_c/2$ ). This gives:

$$p = 4d - 4(2r_c) + 4(\pi r_c/2) \quad (\text{S4})$$

Substitution of Eq. (S1) and simplifying gives:

$$p = 4(1 - [1 - \pi/4]\gamma)d \quad (\text{S5})$$

which is Eq. (5) in the main paper.

### ***S1.2. Calculation of $\gamma$ from Channel Perimeter and Cross-Sectional Area***

Given that it is probably easier to measure the perimeter and cross-sectional area of a coated channel than the radius of curvature from an image of the channel, it is useful to have an equation for calculating  $\gamma$  from  $p$  and  $A_C$ .

Eliminating  $d$  between Eq. (S3) and (S5) gives:

$$A_C = \frac{(1 - [1 - \pi/4]\gamma^2)p^2}{16(1 - [1 - \pi/4]\gamma)^2} \quad (S6)$$

This rearranges to give a quadratic in  $\gamma$ :

$$(16A_C[1 - \pi/4]^2 + p^2[1 - \pi/4])\gamma^2 - 32A_C[1 - \pi/4]\gamma + (16A_C - p^2) = 0 \quad (S7)$$

which can be solved using the usual equation. The required solution is:

$$\gamma = \frac{16A_C[1 - \pi/4] - \sqrt{16^2A_C^2[1 - \pi/4]^2 - (16A_C[1 - \pi/4]^2 + p^2[1 - \pi/4])(16A_C - p^2)}}{16A_C[1 - \pi/4]^2 + p^2[1 - \pi/4]} \quad (S8)$$

This equation enables  $\gamma$  to be determined from the values of  $p$  and  $A_C$  for a square channel with rounded corners (or a square or circular channel).

### ***S1.3. Calculation of Geometric Parameters for Uncoated Monoliths***

Figure S1 depicts the geometry of a coated monolith channel. The width of the uncoated channel,  $d_{UC}$ , is given by:

$$d_{UC} = \frac{1}{\sqrt{\rho_{Cell}}} - w_W \quad (S9)$$

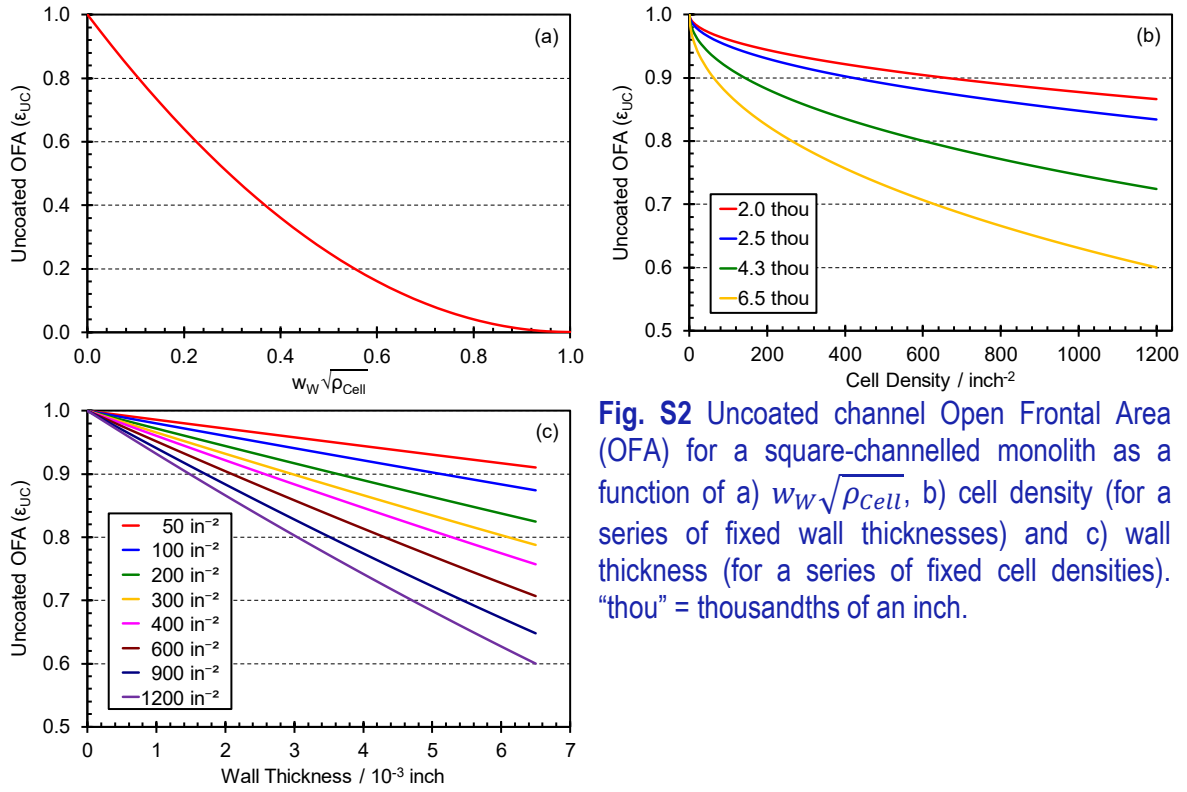
where  $\rho_{Cell}$  is the cell density of the monolith and  $w_W$  is the thickness of the monolith wall. Note that  $1/\sqrt{\rho_{Cell}}$  gives the cell pitch (or cell repeat distance) of the monolith. Obviously, this equation requires that the uncoated channels are square.

The open frontal area (or open area fraction) of the uncoated channel,  $\epsilon_{UC}$ , is given by  $d_{UC}^2 \rho_{Cell}$ . Substituting Eq. (S9) into this gives:

$$\epsilon_{UC} = 1 - w_W \sqrt{\rho_{Cell}} (2 - w_W \sqrt{\rho_{Cell}}) \quad (S10)$$

Thus,  $\epsilon_{UC}$  is a quadratic in  $w_W \sqrt{\rho_{Cell}}$ . Note that  $w_W \sqrt{\rho_{Cell}}$  is the ratio of the wall thickness to the cell pitch; when this is unity, “the wall fills the channel” and  $\epsilon_{UC}$  is zero. This equation applies only to square channels.

Figure S2 shows the variation in  $\epsilon_{UC}$  with  $w_W \sqrt{\rho_{Cell}}$ , cell density (for fixed wall thickness) and wall thickness (for fixed cell density) according to Eq. (S10);  $\epsilon_{UC}$  decreases with increasing cell density and wall thickness.



**Fig. S2** Uncoated channel Open Frontal Area (OFA) for a square-channelled monolith as a function of a)  $w_W \sqrt{\rho_{Cell}}$ , b) cell density (for a series of fixed wall thicknesses) and c) wall thickness (for a series of fixed cell densities). “thou” = thousandths of an inch.

#### S1.4. Calculation of Geometric Parameters for Coated Monoliths

The cross-sectional area occupied by washcoat per channel can be calculated as follows. The ratio of the cross-sectional area of washcoat to the total cross-sectional area of the monolith is  $L_{WC}/\rho_{WC}$ , where  $L_{WC}$  is the washcoat loading by mass (in  $\text{kg m}^{-3}$ ) and  $\rho_{WC}$  is the density of the washcoat after coating. Multiplying this by the cross-sectional area of the monolith divided by the number of channels ( $1/\rho_{Cell}$ ) gives the cross-sectional area occupied by washcoat per channel, as required. The cross-sectional area occupied by washcoat per channel is also given by the difference in cross-sectional area between the uncoated and coated monolith channels, i.e.  $d_{UC}^2 - A_C$ , where  $A_C$  is the cross-sectional area of the open coated channel. Equating these two gives:

$$d_{UC}^2 - A_C = \frac{L_{WC}}{\rho_{WC} \rho_{Cell}} \quad (\text{S11})$$

This equation enables  $A_C$  to be calculated from the washcoat loading. Multiplying this equation by  $\rho_{Cell}$  and rearranging gives:

$$\epsilon_C = \epsilon_{UC} - \frac{L_{WC}}{\rho_{WC}} \quad (\text{S12})$$

where  $\epsilon_C$  is the open frontal area of the coated monolith ( $= A_C \rho_{Cell}$ ) and  $\epsilon_{UC}$  is the open frontal area of the uncoated monolith ( $= d_{UC}^2 \rho_{Cell}$ ).

Substitution of Eq. (S10) into this gives:

$$\epsilon_C = 1 - w_W \sqrt{\rho_{Cell}} (2 - w_W \sqrt{\rho_{Cell}}) - \frac{L_{WC}}{\rho_{WC}} \quad (\text{S13})$$

Note that Eq. (S11) and (S13) apply to any shape of coating in a square channel, while Eq. (S12) applies to any shape of coating in any shape of channel.

Finally, the geometric surface area of the coated monolith is given by  $p\rho_{cell}$ . This equation applies to any shape of coated and uncoated channel.

### ***S1.5. Minimum and Maximum Washcoat Thickness***

As can be seen in Fig. S1a, the minimum thickness of the washcoat,  $w_{Min}$ , is given by:

$$w_{Min} = \frac{d_{UC} - d}{2} \quad (S14)$$

The maximum washcoat thickness ( $w_{Max}$ ) occurs in the corner of the coated channel along the diagonal (Fig. S1a). From Fig. S1b, it can be seen that:

$$w_{Max} = IK = HJ + JK - HI \quad (S15)$$

Substituting in Eq. (S1) and noting that  $HJ = r_c\sqrt{2}$ ,  $JK = w_{Min}\sqrt{2}$  and  $HI = r_c$  (Fig. S1b) gives:

$$w_{Max} = w_{Min}\sqrt{2} + \frac{(\sqrt{2} - 1)d\gamma}{2} \quad (S16)$$

Note that the calculation of  $w_{Min}$  and  $w_{Max}$  applies only to a coated channel that can be described as a square with rounded corners (or a square or circle).

### ***S1.6. Equation of the Channel Wall***

The equation of the channel wall in polar coordinates is derived in the main paper (Eq. (7)), but is repeated here for convenience:

$$r_r = \begin{cases} \frac{d}{2 \cos \theta} & 0 \leq \theta \leq \hat{B}\hat{O}D \\ \frac{d}{2} \left[ (1 - \gamma)\sqrt{2} \cos(\pi/4 - \theta) + \sqrt{\gamma^2 - 2(1 - \gamma)^2 \sin^2(\pi/4 - \theta)} \right] & \hat{B}\hat{O}D \leq \theta \leq \pi/4 \end{cases} \quad (S17)$$

## S2. Methodology

With the exception of the channel geometry (Section 2.1 of the main paper), the methodology used in this work is very similar to that used in previous work by the author [S1] considering flow in the octagonal channels of octo-square asymmetric particulate filters. Thus, the description of methodology here will be kept to the main points; for further details and explanation, the reader is referred to the previous work [S1].

### S2.1. The Velocity Problem

#### S2.1.1. Momentum Balance

In this work, we consider steady-state, fully developed laminar flow of a Newtonian fluid with constant properties ( $C_p$ ,  $\lambda$ ,  $\mu$ ,  $\rho$ ) in a channel of constant cross section. For these assumptions, the (axial) momentum balance is:

$$0 = -\frac{\partial P}{\partial z} + \mu \nabla^2 V \quad (\text{S18})$$

where  $V$  is the local gas velocity along the channel (axial velocity). The terms in this equation can be interpreted as the resultant force applied to the gas per unit volume in the  $z$ -direction due to pressure (first term) and fluid viscosity (second term).

In polar coordinates this becomes:

$$\frac{\partial^2 V}{\partial r^2} + \frac{1}{r} \frac{\partial V}{\partial r} + \frac{1}{r^2} \frac{\partial^2 V}{\partial \theta^2} = \frac{1}{\mu} \frac{\partial P}{\partial z} \quad (\text{S19})$$

Note that since developed flow is assumed, the Laplacian has no term in  $\partial^2 V / \partial z^2$ .

This equation (Eq. (S19)) is solved in the next section using the boundary conditions given in Table S1. The boundary conditions are discussed in more detail elsewhere [S1].

**Table S1** Summary of the boundary conditions used for the velocity and temperature problems.

<i>Velocity problem</i>	
$V(r_r, \theta) = 0$	No-slip at channel wall
$V(r, \theta) = V(r, \theta \pm 2i\pi/N)$	Rotational symmetry
$\left. \frac{\partial V}{\partial \theta} \right _{\theta=i\pi/N} = 0$	Reflectional symmetry
<i>Temperature problem</i>	
$h(T_m - T_r) = \text{Constant}$	Axially constant heat flux
$T(r_r, \theta) = T_r$	Constant wall temperature
$T^*(r, \theta) = T^*(r, \theta \pm 2i\pi/N)$	Rotational symmetry
$\left. \frac{\partial T^*}{\partial \theta} \right _{\theta=i\pi/N} = 0$	Reflectional symmetry



### S2.1.2. Solution of the Momentum Balance

A general solution to equations like Eq. (S19) is given in Ref. [S2]. Adapting this to a channel with N-fold rotational and reflectional symmetry gives a general solution of Eq. (S19):

$$V = \frac{1}{\mu} \frac{\partial P}{\partial z} \left[ \frac{r^2}{4} - \sum_{j=0}^n a_j r^{jN} \cos jN\theta \right] \quad (\text{S20})$$

where  $a_j$  are arbitrary constants. This solution has been previously used for channels with N-fold rotational (and reflectional) symmetry [S1,S3-S6].

The N term in Eq. (S20) is required to fulfil the rotational symmetry boundary condition in Table S1. Potentially, the solution to Eq. (S19) could also contain terms in  $\sin jN\theta$ , in addition to the terms in  $\cos jN\theta$ , but it does not as these would be inconsistent with the reflectional symmetry boundary condition (Table S1). Both of these points are discussed in more detail elsewhere [S1].

It is worth noting that the previous studies on flow in square channels with rounded corners appear *not* to have included the impact of symmetry in their methodology [S7,S8]. While this is not essential, it does mean that i) the whole channel wall, rather than just the unique portion, needs to be considered when determining the arbitrary constants ( $a_j$ ) from the boundary condition, which is numerically inefficient, and ii) constants which are definitely zero because of symmetry may not evaluate to (exactly) zero resulting in numeric error.

Since each term in the summation in Eq. (S20) is individually a general solution of  $\nabla^2 V = 0$ , the number of terms in the summation ( $n$ ), is somewhat arbitrary, but clearly higher values of  $n$  give more “flexibility” in fitting the boundary condition. Selection of an appropriate value of  $n$ , *i.e.* at what point to truncate the series/summation is discussed in Section S3.

To find the values of  $a_j$ , the normal no-slip boundary condition is applied to the channel wall, *i.e.*  $V(r_f, \theta) = 0$ . Thus, at the channel wall, Eq. (S20) becomes:

$$0 = \frac{r_f^2}{4} - \sum_{j=0}^n a_j r_f^{jN} \cos jN\theta \quad (\text{S21})$$

To determine the values of  $a_j$ , it is convenient write this equation in matrix form:

$$\mathbf{S}\mathbf{A} = \mathbf{D} \quad (\text{S22})$$

where:

$$\mathbf{A} = [a_0 \quad a_1 \quad a_2 \quad \dots \quad a_n]^T \quad (\text{S23})$$

$$\mathbf{D} = [r_f^2/4] \quad (\text{S24})$$

$$\mathbf{S} = [1 \quad r_f^N \cos N\theta \quad r_f^{2N} \cos 2N\theta \quad \dots \quad r_f^{nN} \cos nN\theta] \quad (\text{S25})$$

$r_f$  in these equations is given by Eq. (S17). If the boundary condition is evaluated at multiple points on the channel wall, Eq. (S22) can be extended so that each row of  $\mathbf{D}$  and  $\mathbf{S}$  corresponds to a different point on the channel wall. Thus, if the boundary condition is evaluated at  $m$  points, then  $\mathbf{D}$  and  $\mathbf{S}$  will each have  $m$  rows.

The values of  $a_j$  can be determined as the set of values that minimise the sum of the squares of the residuals of Eq. (S22). Thus, as is shown elsewhere [S9], the values of  $a_j$  are given by:

$$\mathbf{A} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \mathbf{D} \quad (\text{S26})$$

This is known as the least squares method [S10,S11]. This method requires that the boundary condition is evaluated at more points than there are unknowns, *i.e.*  $m > n + 1$ . The rows of  $\mathbf{S}$  and  $\mathbf{D}$  were evaluated for 91 (*i.e.*  $m = 91$ ) evenly spaced values of  $\theta$  over the unique part of the channel ( $0 \leq \theta \leq \pi/4$ ). This corresponds to points on the channel wall at  $\frac{1}{2}^\circ$  intervals. Halving the interval to  $\frac{1}{4}^\circ$  (*i.e.*  $m = 181$ ) changed the values of  $F$  calculated from  $V$  by no more than  $\pm 0.0001$ .

The “point matching” method [S3-S6,S11] or “boundary collocation technique” [S6,S11] is an alternative to the least squares method. Here, the number of points where the boundary condition is evaluated equals the number of unknowns (*i.e.*  $m = n + 1$ ), so Eq. (S22) becomes a set of simultaneous equations, which can be solved to give  $a_j$ . The least squares method was used in this work instead of point matching as previous work by the author found that least squares was the superior method [S1].

In the case of a circular channel ( $\gamma = 1$ ),  $V$  is expected not to vary with  $\theta$  so  $a_j$  will be zero for  $j > 0$  (Eq. (S20));  $a_0$  is non-zero since the  $\cos jN\theta$  term in Eq. (S20) is equal to unity for  $j = 0$  and so is independent of  $\theta$ . Setting  $a_j = 0$  for  $j > 0$  results in Eq. (S20) reducing to the analytical solution for flow in a circular channel. However, in this work  $a_j$  for  $j > 0$  was not forced to be zero for circular channels ( $\gamma = 1$ ), but rather evaluated in the same way as for  $\gamma < 1$  (*i.e.* using Eq. (S26) and assuming the symmetry of a square), as a test of the methodology; values determined by this method can then be compared with the analytical solution for a circular channel.

### S2.1.3. Mean Velocity and Friction Factor Calculation

The mean gas velocity,  $V_m$ , is given by:

$$V_m = \frac{\int_0^{\pi/N} \int_0^{r_r} V r dr d\theta}{\int_0^{\pi/N} \int_0^{r_r} r dr d\theta} \quad (S27)$$

Note that it is only necessary to evaluate the integrals over the one eighth of the channel ( $0 \leq \theta \leq \pi/4$ ) because of the symmetry. Substituting Eq. (S20) into Eq. (S27) and evaluating the integral in  $r$  gives:

$$V_m = \frac{2N}{A_c \mu} \frac{\partial P}{\partial z} \int_0^{\pi/N} \left[ \frac{r_r^4}{16} - \sum_{j=0}^n a_j \frac{r_r^{jN+2} \cos jN\theta}{jN+2} \right] d\theta \quad (S28)$$

The integral in this equation was evaluated numerically (after substitution of Eq. (S17)) using the trapezium rule with values of  $\theta$  in the range  $0 \leq \theta \leq \pi/4$  at a  $\frac{1}{2}^\circ$  interval, *i.e.* over 91 equally spaced points. Halving the interval to  $\frac{1}{4}^\circ$  (*i.e.* numeric integration over 181 points), had no effect on the values of  $F$  calculated from  $V_m$  to five decimal places.

Next, an expression for determining the (Fanning) friction factor will be developed. From the definition of the friction factor [S12], it follows that the axial pressure gradient along the channel is given by:

$$-\frac{\partial P}{\partial z} = \frac{f p \rho V_m^2}{2 A_c} \quad (S29)$$

The Reynolds number,  $Re$ , is defined as  $\rho V_m d_H / \mu$  where  $d_H$  is the hydraulic diameter, defined as  $4A_C/p$ . Substituting these into Eq. (S29) and rearranging gives an expression for calculating  $f Re$ , viz.:

$$f Re = -\frac{8A_C^2}{p^2} \cdot \frac{1}{V_m \mu} \frac{\partial P}{\partial z} \quad (S30)$$

where the term to the right of the dot is obtained from Eq. (S28).

In 1-dimensional models of monolith reactors, the viscous losses can be characterised in terms of the viscous loss coefficient,  $F$ , as in Eq. (1) in the main paper. Hence, an expression for determining  $F$  is also required. Applying incompressible flow to Eq. (1), so that  $\partial \rho V^2 / \partial z = 0$ , gives such an expression:

$$F = -A_C \cdot \frac{1}{V_m \mu} \frac{\partial P}{\partial z} \quad (S31)$$

Note that incompressible flow has been applied in this equation as the velocity field was calculated assuming incompressible flow.

Combining Eq. (S30) and (S31) to eliminate the terms to the right of the dot gives:

$$F = \frac{p^2}{8A_C} f Re \quad (S32)$$

From symmetry, it is clear that maximum gas velocity occurs at the centre of the channel. Thus, an expression for the maximum velocity,  $V_{Max}$ , is obtained by substituting  $r = 0$  into Eq. (S20):

$$V_{Max} = -\frac{a_0}{\mu} \frac{\partial P}{\partial z} \quad (S33)$$

#### S2.1.4. Momentum Flux and Kinetic Energy Flux Correction Factors

The momentum flux ( $\alpha$ ) and kinetic energy flux ( $\alpha_E$ ) correction factors are defined [S11] as:

$$\alpha = \frac{\langle V^2 \rangle}{\langle V \rangle^2} = \frac{\langle V^2 \rangle}{V_m^2} \quad (S34)$$

$$\alpha_E = \frac{\langle V^3 \rangle}{\langle V \rangle^3} = \frac{\langle V^3 \rangle}{V_m^3} \quad (S35)$$

Writing Eq. (S34) in integral form gives:

$$\alpha = \frac{1}{V_m^2} \frac{\int_0^{\pi/N} \int_0^{r_f} V^2 r dr d\theta}{\int_0^{\pi/N} \int_0^{r_f} r dr d\theta} \quad (S36)$$

Substituting Eq. (S20) into this and evaluating the integral in  $r$  gives:

$$\alpha = \frac{2N}{A_C} \left( \frac{1}{V_m \mu} \frac{\partial P}{\partial z} \right)^2 \int_0^{\pi/N} \left[ \frac{r_r^6}{96} - \frac{1}{2} \sum_{j=0}^n \frac{a_j r_r^{jN+4} \cos jN\theta}{jN+4} + \sum_{j=0}^n \sum_{k=0}^n \frac{a_j a_k r_r^{[j+k]N+2} \cos jN\theta \cos kN\theta}{[j+k]N+2} \right] d\theta \quad (S37)$$

Applying the same procedure to Eq. (S35) gives:

$$\alpha_E = \frac{2N}{A_C} \left( \frac{1}{V_m \mu} \frac{\partial P}{\partial z} \right)^3 \int_0^{\pi/N} \left[ \frac{r_r^8}{512} - \frac{3}{16} \sum_{j=0}^n \frac{a_j r_r^{jN+6} \cos jN\theta}{jN+6} + \frac{3}{4} \sum_{j=0}^n \sum_{k=0}^n \frac{a_j a_k r_r^{[j+k]N+4} \cos jN\theta \cos kN\theta}{[j+k]N+4} - \sum_{j=0}^n \sum_{k=0}^n \sum_{l=0}^n \frac{a_j a_k a_l r_r^{[j+k+l]N+2} \cos jN\theta \cos kN\theta \cos lN\theta}{[j+k+l]N+2} \right] d\theta \quad (S38)$$

The integrals in Eq. (S37) and (S38) were evaluated numerically (after substitution of Eq. (S17)) using the trapezium rule with values of  $\theta$  in the range  $0 \leq \theta \leq \pi/4$  at a  $1/2^\circ$  interval, *i.e.* over 91 equally spaced points. Halving the interval to  $1/4^\circ$  (*i.e.* numeric integration over 181 points), had no effect on the values of  $\alpha$  and  $\alpha_E$  calculated to at least six decimal places.

## S2.2. The Temperature Problem

### S2.2.1. Energy Balance

In addition to the assumptions already mentioned for the velocity problem (Section S2.1.1), we assume thermally developed flow, no axial heat conduction (or rather that axial heat transport by conduction is negligible compared to that by convection), no viscous dissipation and no heat sources within the fluid. For these assumptions, the fluid energy balance is:

$$0 = -\rho C_p V \frac{\partial T}{\partial z} + \lambda \nabla^2 T \quad (S39)$$

where  $T$  is the local thermodynamic temperature of the gas. The first term in this equation represents heat transport by forced convection along the channel and the second term heat transport by thermal conduction.

This equation will be solved for the combined boundary conditions of (i) axially constant wall heat flux and (ii) constant peripheral wall temperature, which is the “H1” boundary condition of Shah and London [S11]; these boundary conditions will be applied later in the derivation. The full set of boundary conditions is summarised in Table S1.

Eliminating,  $(\partial P / \partial z) / \mu$  between Eq. (S20) and (S30) and substituting the result into Eq. (S39) gives:

$$0 = \rho C_p \frac{f Re p^2 V_m}{8 A_C^2} \left[ \frac{r^2}{4} - \sum_{j=0}^n a_j r^{jN} \cos jN\theta \right] \frac{\partial T}{\partial z} + \lambda \nabla^2 T \quad (S40)$$

For thermally developed flow and the first thermal boundary condition (axially constant wall heat flux), it can be shown that [S1]:

$$\frac{\partial T}{\partial z} = \frac{\partial T_m}{\partial z} \quad (S41)$$

One way of looking at Eq. (S41) is that axially constant heat flux at the channel wall requires that the derivative of the temperature at the channel wall in the direction normal to the wall (which is proportional to the heat flux at the wall) must be independent of  $z$ . This suggests a temperature solution of the form  $T = g(z) + \Psi(r, \theta)$ , *i.e.* where the part of the solution dependant on  $r$  and  $\theta$  is independent of  $z$ . Here  $g(z)$  is a function of  $z$  only and  $\Psi(r, \theta)$  a function of  $r$  and  $\theta$ . For this solution, it is clear that  $\partial T / \partial z$  is independent of  $r$  and  $\theta$  and hence that Eq. (S41) is correct. This is discussed in more detail elsewhere [S1].

Substituting Eq. (2) (from the main paper) and (S41) into Eq. (S40) to eliminate  $\partial T / \partial z$  gives:

$$\frac{\lambda d_H^3}{8 f Re h (T_m - T_r)} \nabla^2 T = \frac{r^2}{4} - \sum_{j=0}^n a_j r^{jN} \cos jN\theta \quad (S42)$$

where  $d_H (= 4A_C/p)$  is the hydraulic diameter of the channel.

The normalised temperature<sup>2</sup>,  $T^*$ , is defined as:

$$T^* = \frac{\lambda d_H^3}{8 f Re h} \cdot \frac{T - T_r}{T_m - T_r} \quad (S43)$$

Rearranging this to give  $T$  and substituting the result into Eq. (S42) gives:

$$\nabla^2 T^* = \frac{r^2}{4} - \sum_{j=0}^n a_j r^{jN} \cos jN\theta \quad (S44)$$

Since no axial heat conduction in the gas has been assumed, the Laplacian in this equation has no term in  $\partial^2 T^* / \partial z^2$ . This means that the  $T_r$  term in the numerator of Eq. (S43) has no impact on Eq. (S44) as it goes to zero when operated on by the Laplacian, but it is included as it simplifies determination of the arbitrary constants in the solution of Eq. (S44) from the boundary condition (Section S2.2.2).

Expressing the Laplacian in polar coordinates, Eq. (S44) becomes:

$$\frac{\partial^2 T^*}{\partial r^2} + \frac{1}{r} \frac{\partial T^*}{\partial r} + \frac{1}{r^2} \frac{\partial^2 T^*}{\partial \theta^2} = \frac{r^2}{4} - \sum_{j=0}^n a_j r^{jN} \cos jN\theta \quad (S45)$$

### S2.2.2. Solution of the Energy Balance

For a channel with  $N$ -fold rotational and reflectional symmetry, a general solution of Eq. (S45) is [S1,S3,S5,S10]:

---

<sup>2</sup> Since  $T^*$  has units ( $m^4$ ), it is referred to as “normalised” rather than “non-dimensionalised”.

$$T^* = \frac{r^4}{64} - \sum_{j=0}^n \frac{a_j r^{jN+2} \cos jN\theta}{4(jN+1)} - \sum_{j=0}^n c_j r^{jN} \cos jN\theta \quad (\text{S46})$$

As with the velocity solution (Eq. (S20), Section S2.1.2), the  $N$  in this equation is required to fulfil the rotational symmetry boundary condition (Table S1) and the equation has no terms in  $\sin jN\theta$  to fulfil the reflectional symmetry boundary condition (Table S1).

To find the values of  $c_j$ , the second thermal boundary condition (constant wall temperature) is applied, *i.e.*  $T(r_r, \theta) = T_r$ , where  $T_r$  is the temperature at the channel wall, a constant. Substituting this into Eq. (S43) gives  $T^*(r_r, \theta) = 0$ . Thus, at the channel wall, Eq. (S46) becomes:

$$0 = \frac{r_r^4}{64} - \sum_{j=0}^n \frac{a_j r_r^{jN+2} \cos jN\theta}{4(jN+1)} - \sum_{j=0}^n c_j r_r^{jN} \cos jN\theta \quad (\text{S47})$$

To determine the values of  $c_j$ , it is convenient to write this equation in matrix form, in a similar way to that used for determining  $a_j$  in Section S2.1.2:

$$\mathbf{SC} = \mathbf{E} - \mathbf{UA} \quad (\text{S48})$$

where:

$$\mathbf{C} = [c_0 \quad c_1 \quad c_2 \quad \dots \quad c_n]^T \quad (\text{S49})$$

$$\mathbf{E} = [r_r^4/64] \quad (\text{S50})$$

$$\mathbf{U} = \left[ \frac{r_r^2}{4} \quad \frac{r_r^{N+2} \cos N\theta}{4(N+1)} \quad \frac{r_r^{2N+2} \cos 2N\theta}{4(2N+1)} \quad \dots \quad \frac{r_r^{nN+2} \cos nN\theta}{4(nN+1)} \right] \quad (\text{S51})$$

As was done for the velocity problem (Section S2.1.2), Eq. (S48) can be extended, such that if the boundary condition is evaluated at  $m$  points on the channel wall,  $\mathbf{E}$ ,  $\mathbf{S}$  and  $\mathbf{U}$  will have  $m$  rows with each row corresponding to a different point on the channel wall. The values of  $c_j$  (*i.e.* the elements of  $\mathbf{C}$ ) can then be obtained using the least squares method:

$$\mathbf{C} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T (\mathbf{E} - \mathbf{UA}) \quad (\text{S52})$$

For a circular channel ( $\gamma = 1$ ),  $T^*$ , like  $V$  (Section S2.1.2), is expected not to vary with  $\theta$  so  $a_j$  and  $c_j$  will be zero for  $j > 0$  (Eq. (S46)). Setting  $a_j = c_j = 0$  for  $j > 0$  results in Eq. (S46) reducing to the analytical solution for a circular channel. As with the velocity problem,  $a_j$  and  $c_j$  were not forced to be zero for circular channels for  $j > 0$ , but rather evaluated in the same way as for  $\gamma < 1$  (*i.e.* using Eq. (S52)), as a test of the methodology.

### S2.2.3. Nusselt Number Calculation

The normalised mixing cup temperature of the gas in the channel,  $T_m^*$ , is given by:

$$T_m^* = \frac{2N}{A_C} \int_0^{\pi/N} \int_0^{r_r} \frac{V}{V_m} T^* r dr d\theta \quad (\text{S53})$$

Substituting Eq. (S20) and (S46) into this and evaluating the integral in  $r$  gives:

$$\begin{aligned}
T_m^* = \frac{2N}{A_c} \left( \frac{1}{V_m \mu} \frac{\partial P}{\partial z} \right) \int_0^{\pi/N} & \left[ \frac{r_r^8}{2048} - \frac{1}{64} \sum_{j=0}^n \frac{a_j (jN+5) r_r^{jN+6} \cos jN\theta}{(jN+1)(jN+6)} \right. \\
& - \frac{1}{4} \sum_{j=0}^n \frac{c_j r_r^{jN+4} \cos jN\theta}{jN+4} \\
& + \sum_{j=0}^n \sum_{k=0}^n a_j \left[ \frac{a_k r_r^{[j+k]N+4}}{4(jN+1)([j+k]N+4)} \right. \\
& \left. \left. + \frac{c_k r_r^{[j+k]N+2}}{[j+k]N+2} \right] \cos jN\theta \cos kN\theta \right] d\theta
\end{aligned} \tag{S54}$$

This integral was evaluated numerically (after substitution of Eq. (S17)) using the trapezium rule with values of  $\theta$  in the range  $0 \leq \theta \leq \pi/4$  at a  $1/2^\circ$  interval, as was done when determining  $V_m$ ,  $\alpha$  and  $\alpha_E$  (Section S2.1.3 and S2.1.4). Halving the interval to  $1/4^\circ$  had no effect on the values of  $Nu$  calculated from  $T_m^*$  to at least seven decimal places.

From Eq. (S43), it follows that  $T_m^*$  is also given by:

$$T_m^* = \frac{\lambda d_H^3}{8f Re h} \tag{S55}$$

Rearranging this gives an expression for the Nusselt number,  $Nu$ , viz.:

$$Nu = \frac{h d_H}{\lambda} = \frac{d_H^4}{8(f Re) T_m^*} \tag{S56}$$

Thus,  $Nu$  is calculated by eliminating  $T_m^*$  between Eq. (S54) and (S56).

The energy balance of a 1-dimensional model of a monolith reactor contains  $ph$ , not  $Nu$  (Eq. (2) in the main paper). It is therefore useful to have an expression for this. Combining the definitions of  $Nu$  ( $= h d_H / \lambda$ ) and  $d_H$  ( $= 4A_c / p$ ) with  $ph$  gives such an expression:

$$\frac{ph}{\lambda} = \frac{p^2}{4A_c} Nu \tag{S57}$$

## S2.3. References

- S1. Watling, T. C. Flow and Forced Convection Heat Transfer Characteristics of Developed Laminar Flow in the Octahedral Channels of Octo-Square Asymmetric Particulate Filters. *Res. in Eng.* **2020**, 5, 100086, [doi:10.1016/j.rineng.2019.100086](https://doi.org/10.1016/j.rineng.2019.100086).
- S2. Albright, L.F. (Ed.). *Albright's Chemical Engineering Handbook*; CRC Press: Boca Raton, 2009, pp. 128-130, [doi:10.1201/9781420014389](https://doi.org/10.1201/9781420014389).
- S3. Cheng, K.C. Laminar Flow and Heat Transfer Characteristics in Regular Polygonal Ducts. Proceedings of Third International Heat Transfer Conference (7<sup>th</sup>-12<sup>th</sup> August, Chicago, 1966) 1, 64-76, [doi:10.1615/IHTC3.1590](https://doi.org/10.1615/IHTC3.1590).
- S4. Cheng, K.C. Dirichlet Problem for Laminar Forced Convection with Heat Sources and Viscous Dissipation in Regular Polygonal Ducts. *AIChE J.* **1967**, 13, 1175-1180, [doi:10.1002/aic.690130626](https://doi.org/10.1002/aic.690130626).

- S5. Cheng, K.C. Laminar Forced Convection in Regular Polygonal Ducts With Uniform Peripheral Heat Flux. *J Heat Trans.* **1969**, 91, 156-157, [doi:10.1115/1.3580075](https://doi.org/10.1115/1.3580075).
- S6. Shih, F.S. Laminar Flow in Axisymmetric Conduits by a Rational Approach. *Can. J. Chem. Eng.* **1967**, 45, 285-294, [doi:10.1002/cjce.5450450507](https://doi.org/10.1002/cjce.5450450507).
- S7. Ray, S.; Misra, D. Laminar fully developed flow through square and equilateral triangular ducts with rounded corners subjected to H1 and H2 boundary conditions. *Int. J. Therm. Sci.* **2010**, 49, 1763-1775, [doi:10.1016/j.ijthermalsci.2010.03.012](https://doi.org/10.1016/j.ijthermalsci.2010.03.012).
- S8. Lorenzi, M.; Morini, G.L. Single-Phase Laminar Forced Convection in Microchannels With Rounded Corners. *Heat Trans. Eng.* **2011**, 32, 1108-1116, [doi:10.1080/01457632.2011.562457](https://doi.org/10.1080/01457632.2011.562457).
- S9. Etheridge, J.E.; John, G.; Watling, T.C. Application of Surrogate Modelling to the Optimisation of Kinetic Parameters in an Emissions Control Catalyst Model Using Vehicle Drive Cycle Data. *Emiss. Control Sci. Technol.* **2017**, 3, 310-322, [doi:10.1007/s40825-017-0069-z](https://doi.org/10.1007/s40825-017-0069-z).
- S10. Shah, R.K. Laminar Flow Friction and Forced Convection Heat Transfer in Ducts of Arbitrary Geometry. *Int J. Heat Mass Transfer* **1975**, 18, 849-862, [doi:10.1016/0017-9310\(75\)90176-3](https://doi.org/10.1016/0017-9310(75)90176-3).
- S11. Shah, R. K.; London, A.L. *Laminar Flow Forced Convection in Ducts*; Academic Press: New York, 1978, pp. 16-31, 43, 65-66, 78, 82-83, 198, 200, [doi:10.1016/C2013-0-06152-X](https://doi.org/10.1016/C2013-0-06152-X).
- S12. Bird, R.B.; Stewart, W.E.; Lightfoot, E.N. *Transport Phenomena, Revised Second Edition*. John Wiley & Sons: New York, 2007, pp. 178-179, ISBN: 978-0-470-11539-8.



### S3. Convergence Behaviour

It is necessary to understand the convergence behaviour of Eq. (S20) and (S46) to decide at what point to truncate the series, *i.e.* what value of  $n$  should be used. Tables S2 and S3 show how the values of  $F$  and  $Nu$  calculated for different values of  $\gamma$  vary with  $n$ . Increasing  $n$  results in convergence of the calculated values, although the change is small, being confined to the fifth significant figure for both  $F$  and  $Nu$ . In general, the maximum deviation of the velocity and normalised temperature at the channel wall from the intended boundary value of zero decrease with increasing  $n$  (not shown).  $n = 7$  was used for calculating the values in Table S4 and Fig. 2, 3 and 5-10 (in the main paper) as this results in convergence to the required precision.

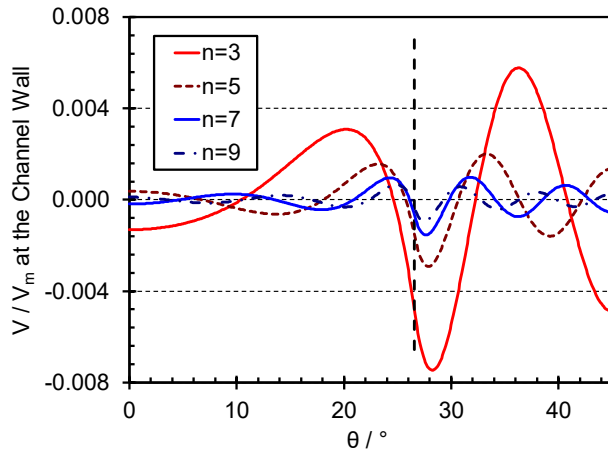
**Table S2** Effect of  $n$  on the value of  $F$  calculated for different  $\gamma$ .

<b>n</b>	<b>Viscous Loss Coefficient, F</b>								
	<b><math>\gamma</math>:</b>	<b>0</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>	<b>0.4</b>	<b>0.6</b>	<b>0.8</b>	<b>1.0</b>
3		28.453	28.338	28.026	27.584	27.080	26.088	25.379	25.133
5		28.454	28.338	28.025	27.585	27.080	26.090	25.378	25.133
7		28.454	28.338	28.025	27.585	27.081	26.090	25.378	25.133
9		28.454	28.338	28.026	27.585	27.081	26.090	25.378	25.133

**Table S3** Effect of  $n$  on the value of  $Nu$  calculated for different  $\gamma$ .

<b>n</b>	<b>Nusselt Number, Nu</b>								
	<b><math>\gamma</math>:</b>	<b>0</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>	<b>0.4</b>	<b>0.6</b>	<b>0.8</b>	<b>1.0</b>
3		3.6082	3.7590	3.8974	4.0171	4.1153	4.2527	4.3318	4.3636
5		3.6080	3.7590	3.8974	4.0170	4.1153	4.2528	4.3319	4.3636
7		3.6080	3.7590	3.8974	4.0170	4.1152	4.2528	4.3319	4.3636
9		3.6080	3.7590	3.8974	4.0170	4.1152	4.2528	4.3319	4.3636

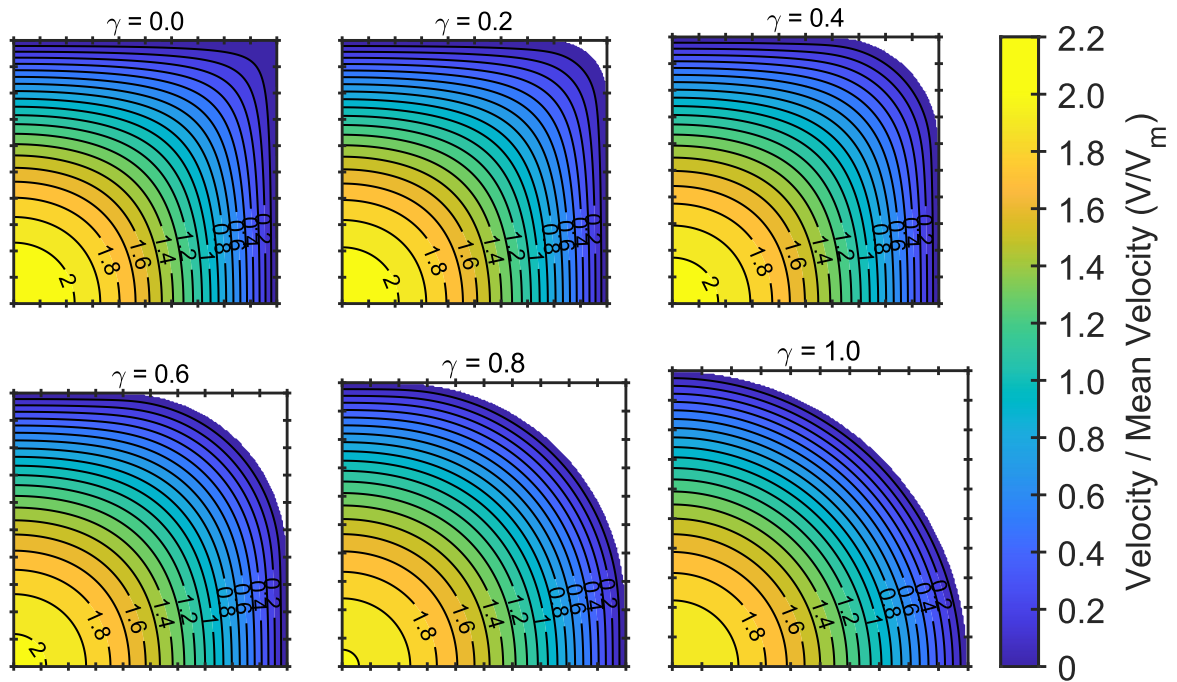
The deviation of the predicted gas velocity at the channel wall from the intended boundary value of zero is shown in Fig. S3 for  $\gamma=1/2$  for four values of  $n$  ( $n=3, 5, 7, 9$ ). The dashed vertical line in this figure indicates the angle where the straight and round sides meet. Deviations from zero decrease markedly as  $n$  is increased. Deviations from zero are larger for the round side ( $26.6^\circ \leq \theta \leq 45^\circ$ ) than the straight side ( $0^\circ \leq \theta \leq 26.6^\circ$ ); for the straight side deviations from zero tend to be larger in the region close to the round side than in the centre of the straight side ( $\theta=0^\circ$ ).



**Fig. S3** Deviation of predicted velocity at the channel wall from zero for  $\gamma=1/2$ . The vertical line indicates the angle where the straight and round sides meet.

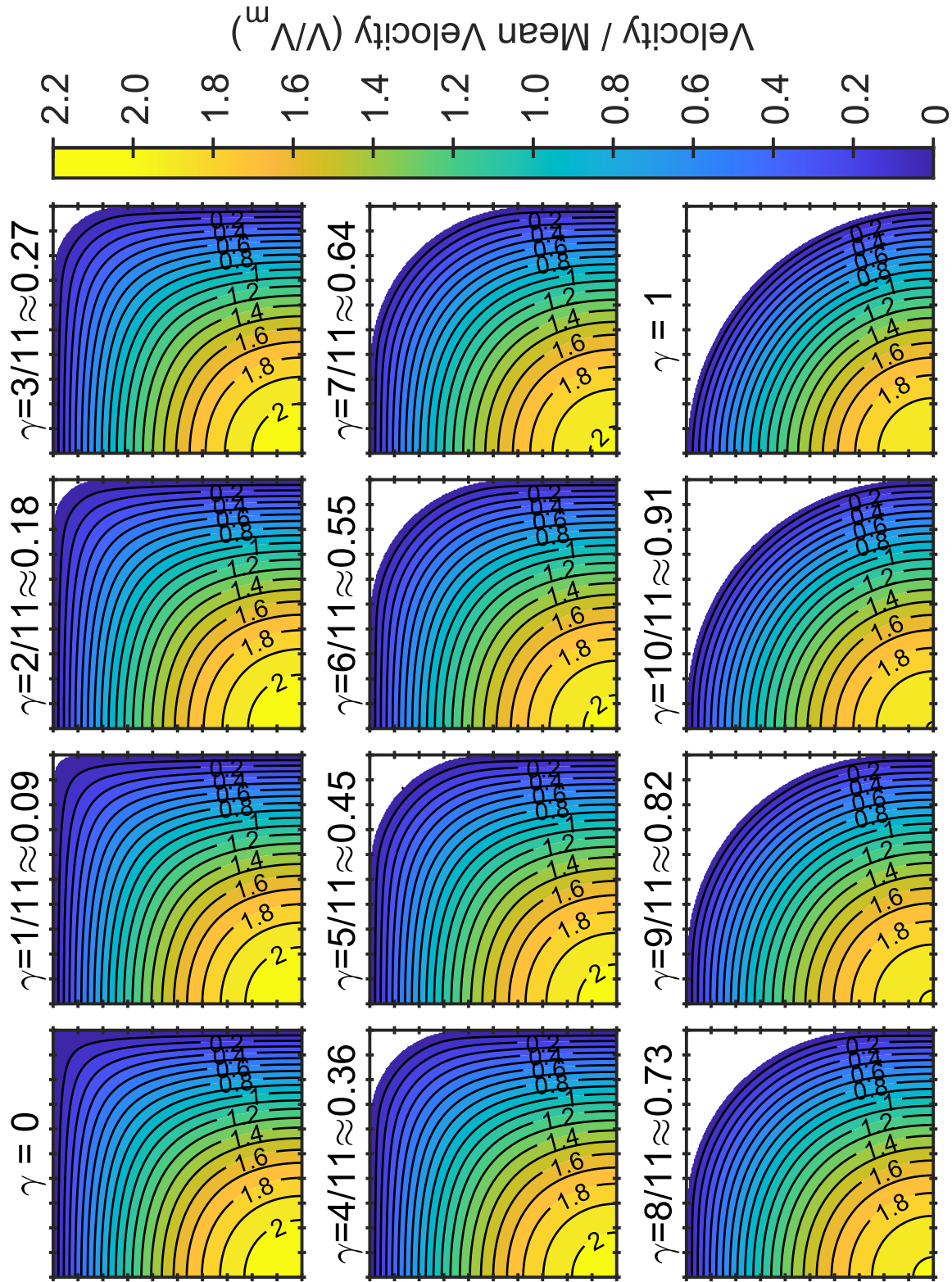
## S4. The Velocity and Temperature Field

This section of the Supplementary Material contains more contour plots. Figure S4 shows the same data as Fig. 2 in the main paper but is plotted such that each “channel” has the same cross-sectional area; this is equivalent to a comparison of different channel shapes with the same substrate at constant washcoat loading. Figure S5 shows velocity contour plots for more values of  $\gamma$  than shown in the main paper. Figure S6 shows contour plots of the non-dimensionalised temperature. The temperature field resembles the velocity field in terms of the shape of the contours, but the numbers are different. The figures are stored as vector graphics and so can be enlarged to any size in a PDF viewer.

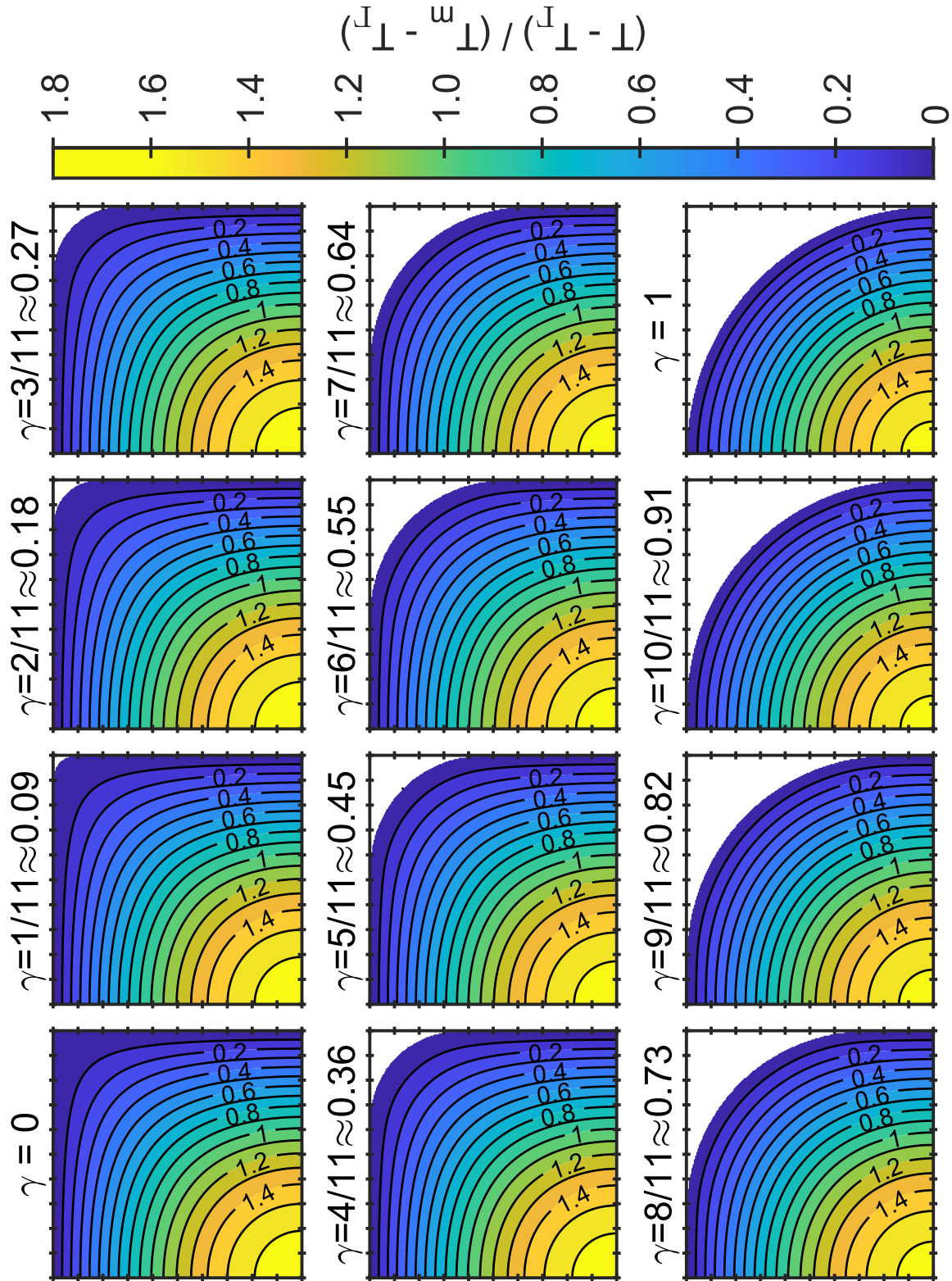


**Fig. S4** Velocity contour plots for developed laminar flow in channels going from a square ( $\gamma=0$ ) to a circle ( $\gamma=1$ ) plotted such that all channels have the same cross-sectional area. This is the same data as shown in Fig. 2 of the main paper. Only one quadrant of the channel is shown. Velocity field calculated with  $n=7$ ,  $m=91$ .

**Fig. S5** Velocity contour plots for developed laminar flow in channels going from square ( $\gamma=0$ ) to a circle ( $\gamma=1$ ). Only one quadrant of the channel is shown. Velocity field calculated using the least squares method with  $n=7$ ,  $m=91$ .



**Fig. S6** Non-dimensionalised temperature contour plots for fully developed laminar flow in channels going from square ( $\gamma=0$ ) to a circle ( $\gamma=1$ ). Only one quadrant of the channel is shown. Temperature field calculated using the least squares method with  $n=7$ ,  $m=91$ .



## S5. Tabulated Data Characterising Fully Developed Laminar Flow

**Table S4** Flow and heat transfer characteristics for fully developed laminar flow as a function of channel geometry going from a square channel ( $\gamma=0$ ) to a circle ( $\gamma=1$ ). The maximum deviation of the velocity ( $V_{\Gamma,Max}/V_m$ ) and normalised temperature ( $T_{\Gamma,Max}^*/T_m^*$ ) at the channel wall from the intended boundary values of zero are also given as a measure of accuracy. Values calculated with  $n=7$ ,  $m=91$ .

$\gamma$	F	f Re	$\alpha$	$\alpha_E$	$\frac{V_{max}}{V_m}$	Nu	$\frac{ph}{\lambda}$	$\frac{V_{\Gamma,max}}{V_m} / \%$	$\frac{T_{\Gamma,max}^*}{T_m^*} / \%$
0.000	28.454	14.227	1.378	2.154	2.096	3.608	14.432	0.05	0.00
0.025	28.447	14.375	1.378	2.154	2.096	3.646	14.432	0.04	0.01
0.050	28.424	14.514	1.378	2.152	2.095	3.685	14.431	0.12	0.03
0.075	28.388	14.644	1.377	2.149	2.094	3.722	14.431	0.16	0.04
0.100	28.338	14.765	1.376	2.146	2.092	3.759	14.429	0.13	0.04
0.125	28.275	14.878	1.374	2.142	2.090	3.795	14.425	0.10	0.03
0.150	28.202	14.982	1.373	2.137	2.087	3.830	14.420	0.12	0.04
0.175	28.118	15.078	1.371	2.131	2.085	3.864	14.413	0.16	0.06
0.200	28.025	15.166	1.370	2.126	2.081	3.897	14.404	0.17	0.07
0.225	27.925	15.248	1.368	2.120	2.078	3.929	14.392	0.16	0.06
0.250	27.817	15.322	1.366	2.113	2.075	3.960	14.378	0.14	0.05
0.275	27.704	15.390	1.364	2.107	2.071	3.989	14.362	0.12	0.05
0.300	27.585	15.452	1.362	2.101	2.067	4.017	14.343	0.13	0.06
0.325	27.463	15.508	1.360	2.094	2.064	4.044	14.321	0.16	0.07
0.350	27.337	15.559	1.359	2.088	2.060	4.069	14.298	0.17	0.08
0.375	27.210	15.605	1.357	2.082	2.056	4.093	14.273	0.16	0.07
0.400	27.081	15.646	1.355	2.076	2.052	4.115	14.245	0.14	0.07
0.425	26.951	15.684	1.353	2.070	2.049	4.137	14.217	0.12	0.06
0.450	26.822	15.717	1.352	2.064	2.045	4.157	14.186	0.12	0.06
0.475	26.694	15.748	1.350	2.059	2.041	4.175	14.155	0.14	0.07
0.500	26.567	15.775	1.349	2.054	2.038	4.193	14.123	0.15	0.08
0.525	26.443	15.799	1.347	2.049	2.034	4.209	14.091	0.15	0.08
0.550	26.322	15.821	1.346	2.044	2.031	4.225	14.058	0.13	0.07
0.575	26.204	15.840	1.344	2.039	2.028	4.239	14.026	0.11	0.06
0.600	26.090	15.858	1.343	2.035	2.025	4.253	13.994	0.10	0.06
0.625	25.981	15.874	1.342	2.031	2.022	4.265	13.962	0.12	0.06
0.650	25.876	15.889	1.341	2.027	2.019	4.277	13.932	0.13	0.07
0.675	25.777	15.902	1.340	2.023	2.017	4.288	13.902	0.14	0.07
0.700	25.684	15.914	1.339	2.020	2.014	4.298	13.874	0.12	0.07
0.725	25.598	15.925	1.338	2.017	2.012	4.308	13.848	0.11	0.06
0.750	25.517	15.935	1.337	2.014	2.010	4.316	13.824	0.08	0.05
0.775	25.444	15.945	1.337	2.012	2.008	4.324	13.801	0.09	0.05
0.800	25.378	15.954	1.336	2.009	2.007	4.332	13.781	0.11	0.06
0.825	25.320	15.963	1.335	2.007	2.005	4.339	13.764	0.12	0.07
0.850	25.269	15.971	1.335	2.005	2.004	4.345	13.748	0.11	0.06
0.875	25.227	15.978	1.334	2.004	2.003	4.350	13.736	0.09	0.05
0.900	25.192	15.985	1.334	2.002	2.002	4.355	13.726	0.06	0.03
0.925	25.166	15.991	1.334	2.001	2.001	4.358	13.718	0.09	0.05
0.950	25.147	15.996	1.334	2.001	2.000	4.361	13.713	0.11	0.06
0.975	25.136	15.999	1.333	2.000	2.000	4.363	13.710	0.11	0.06
1.000	25.133	16.000	1.333	2.000	2.000	4.364	13.709	0.00	0.00



## S6. MATLAB® Program

The MATLAB® program below was used in this work for calculating the velocity and temperature fields, producing the tabulated values found in Table S4 and producing the contour plots appearing in this work. This program was based on one used in a previous study [S1].

In the program,  $d$  is taken to be unity and hence all quantities depending on  $d$  ( $A_C$ ,  $a_j$ ,  $c_j$ ,  $d_H$ ,  $p$ ,  $r$  and  $r_T$ ) are relative to  $d$ ; this has no impact on the reported values as they are all dimensionless. Thus, for the purposes of the program/computation, the units of these quantities are different from that given in the nomenclature.

The obvious implementation of Eq. (S26) and (S52) in MATLAB® uses left divide [S13], viz.:

$$\begin{aligned} A &= (S' * S) \setminus S' * D; \\ C &= (S' * S) \setminus S' * (E - U * A); \end{aligned}$$

However, with high values of  $n$ , MATLAB® warns that  $S$  is badly scaled or nearly singular. This is because  $r_T^{jN} \cos jN\theta$ , which forms the elements of  $S$ , decreases dramatically with increasing  $j$  and that higher values of  $j$  are encountered as  $n$  increases.

MATLAB®'s `pinv` [S14] avoids this problem by treating elements of  $S$  that are smaller than a specific tolerance as zero [S14]. Thus, the following code was used:

$$\begin{aligned} A &= \text{pinv}(S) * D; \\ C &= \text{pinv}(S) * (E - U * A); \end{aligned}$$

Using `pinv` instead of left divide had no impact on the calculated values ( $F$ ,  $fRe$ ,  $Nu$ ,  $V_{Max}/V_m$ ,  $\alpha$ ,  $\alpha_E$ ) to at least 10 decimal places, but could have a significant effect on values of  $a_j$  and  $c_j$  obtained for higher  $j$  (which indicates that the effected values of  $a_j$  and  $c_j$  have negligible contribution to  $V$  and  $T^*$ ).

```
function out = SquareRoundedCornersChannelFlow
% Runs calculation for a series of different values of gamma. Also makes velocity and temperature contour plots
% Columns of out: gamma, F, fRe, alpha, alphaE, Vmax/Vm, Nu, p*h/lambda, VwMax/Vm as %, TwMax/Tm as %, p^2/Ac, p^2*d^2/Ac^2, d/sqrt(Ac)
% Code written in MATLAB R2020a. Using an earlier version may cause problems

%-Parameters controlling calculation-----
gValues = 0:0.2:1; % Array of values of gamma for which calculation is to be run
```

---

S13. MathWorks®, “mldivide, \: Solve systems of linear equations  $Ax = B$  for  $x$ ”, <https://uk.mathworks.com/help/matlab/ref/mldivide.html>, accessed August 2020.

S14. MathWorks®, “pinv: Moore-Penrose pseudoinverse”, <https://uk.mathworks.com/help/matlab/ref/pinv.html>, accessed August 2020.

```

% gValues = 0:0.025:1; % Array of values of gamma for which calculation is to be run

n      = 7 ; % Number of terms in algebraic-trigonometric series (excluding zero term)
m      = 91 ; % Number of points at which boundary conditions are evaluated. Must have m>n+1
mAng   = 91 ; % Number of angles used for numeric integration to obtain Vm, alpha, alphaE and Nu
mG     = 200 ; % Number of points along side of Cartesian grid for velocity and temperature contour plots

% Parameters controlling if plots made, whether V or T plot made and number of columns of subplots
Plt     = true ; % Creates profile plots if true
vPlt    = true ; % Creates velocity plot if true, temperature plot if false (provided Plt true)
nCol    = 3 ; % Number of columns for arranging plots
%-----

d       = 1 ; % Width of channel, arbitrary units. As all calculated values dimensionless, dimensions can be in arbitrary units,
provided they are consistent
nGamma = numel(gValues) ; % Number of values of gamma for running calculation

% Parameters controlling plotting
FntSz   = 12 ; % Font size
FntSzC  = FntSz*2/3 ; % Font size for contour line labels
Gap     = 0.1 ; % Fraction of space allocated to (subplot+gap) that becomes the gap
adjT    = 1.1 ; % Title allowance/adjustment - how much extra height to allow for subplot title as fraction of subplot height

% Calculated parameters controlling plotting
nRow    = ceil(nGamma/nCol); % Number of rows for arranging plots
wCol    = 1/(nCol+0.66) ; % Width of frame for each subplot or for colour bar. Assumes axis units set to 'normalized' so coordinates between 0
and 1. NB Colour bar needs less space than plot, hence 0.66
Edge    = wCol*Gap ; % Fractional distance between subplots and left & top of figure
Width   = wCol*(1-Gap) ; % Width of subplot

% Loop through requested values of gamma, running calculation and compiling results. Create plots if Plt is true
out = zeros(nGamma,13);
col = 0;
row = 1;
for i = 1:nGamma
    gamma = gValues(i);
    [values,x,y,T,V] = ChannelCalc(d,gamma,n,m,mAng,mG);
    out(i,:) = values;

    % Make plots, if required
    if Plt
        % Calculate row and column for subplot
        col = col + 1;
        if col > nCol
            row = row + 1;
            col = 1 ;
        end

        ax = subplot(nRow,nCol,i);
    end
end

```

```

if i==1 % Only need to calculate these values once
    Pos = ax.Parent.Position; % Position and size of figure
    aRatio = Pos(4)/Pos(3) ; % Aspect ratio of figure, height/width
    Ht = Width/aRatio ; % Height of subfigure. Corrects for aspect ratio of figure, so subplot is square
    hRow = wCol /aRatio ; % Height of frame for each subplot
end

if vPlt % Make velocity plot
    cLabel = 'Velocity / Mean Velocity (V/V_m)' ; % Label for colour bar
    tLabels = {'0','0.2','0.4','0.6','0.8','1.0','1.2','1.4','1.6','1.8','2.0','2.2'}; % Tick labels for colour bar
    z = V ; % Quantity to be plotted, V, actually V/Vm
else % Make temperature plot
    cLabel = '(T - T_\Gamma) / (T_m - T_\Gamma)' ; % Label for colour bar
    tLabels = {'0','0.2','0.4','0.6','0.8','1.0','1.2','1.4','1.6','1.8'}; % Tick labels for colour bar
    z = T ; % Quantity to be plotted, T, actually non-dimensionalised
temperature
end

% Make contour plot. Verify no plotted values exceed maximum on colour bar
Max = str2double(tLabels(end)); % Maximum value on colour bar
[c,h] = contourf(ax,x,y,z,'LabelSpacing',500,'LevelStep',0.1,'ShowText','on','TextList',0.2:0.2:Max);
if max(z,[],'all')>Max, error('Maximum on colour bar is less than maximum value of V/Vm or T/Tm.');
```

```

end

% Format subplot, add title, move & resize subplot
set(ax,'LineWidth',1,'TickDir','both','XTickLabel',[],'YTickLabel',[],'XTick',linspace(0,d/2,11),'YTick',linspace(0,d/2,11),'Zlim',[0,Max]); % Format subplot axes
clabel(c,h,'FontSize',FntSzC) ; % Format contour labels
title(ax,['\gamma = ',sprintf('%1f',gamma)],'FontSize',FntSz,'FontWeight','normal'); % Add title
ax.Position = [wCol*(col-1)+Edge,1-(hRow*(row-1)+Ht)*adjT-Edge/aRatio,Width,Ht] ; % Move and resize plot

% If last subplot, add colour bar
if i==nGamma, AddColourBar(ax,FntSz,cLabel,tLabels,Max,wCol*nCol+Edge); end
end % Plots
end % Loop on gamma values
end

%-----
function [values,x,y,T,V] = ChannelCalc(d,gamma,n,m,mAng,mG)
% Calculation, for a single set of parameters, of flow profile, friction factor, Nu, etc for flow a square channel with rounded corners

% Constant
N = 4; % Rotational symmetry of channel

% Geometric calculations
[Ac,BOD,dH,p2_divAc] = GeoCalc(d,gamma);

% Determine constants in algebraic-trigonometric series from boundary conditions
[A,C] = EvalConstants(BOD,d,gamma,n,N,m);

```



```

% Write A and C to Command Window
WriteConsts(gamma,A,'A');
WriteConsts(gamma,C,'C');
fprintf('\n');

% Determine momentum and kinetic energy flux correction factors, mean velocity & normalised mixing cup temperature
[alpha,alphaE,Tsm,Vm] = VelocityTempCalc(A,Ac,BOD,C,d,gamma,mAng,N,n);

% Determine velocity & temperature profiles
[T,V,x,y] = VelocityTempField(A,BOD,C,d,gamma,mG,N,n,Tsm,Vm);

% Evaluate largest deviation of velocity & temperature at channel edge from BC of V=0 and T*=0 as measure of accuracy of V & T solution
[TwMax,VwMax] = BoundaryCheck(A,BOD,C,d,gamma,N,n,Tsm,Vm);

% Calculate F, fRe, Nu and (p*h/lambda)
F = Ac/Vm ; % NB Vm is actually (mean velocity)(viscosity)/(-dP/dz)
fRe = 8*F/p2_divAc ;
Nu = dH^4/(8*fRe*Tsm);
ph_divLamda = p2_divAc*Nu/4 ;

% Compile numeric output into an array
values = [gamma,F,fRe,alpha,alphaE,A(1)/Vm,Nu,ph_divLamda,VwMax*100,TwMax*100,p2_divAc,p2_divAc*d^2/Ac,d/sqrt(Ac)];

end

%-----
function AddColourBar(ax,FntSz,Text,tLabels,Max,xBar)
% Adds colour bar, moves it to the right of the subplots and resizes to occupy the full height of the used part of the figure

c = colorbar(ax,'FontSize',FntSz,'Limits',[0,Max],'TickDirection','Both','TickLabels',tLabels,'Ticks',0:0.2:Max);
br = ax.Position; % Position of bottom right plot. Arguments are [left, bottom, width, height]
tl = ax.Parent.Children(end).Position; % Position of top left plot
cP = c.Position; % Position of colour bar
c.Position = [xBar, br(2), 2*cP(3), tl(2)+tl(4)-br(2)]; % Change position of colour bar, so to right of plots and full height
c.LineWidth = 1 ;
c.Label.String = Text ;
c.Label.FontSize = FntSz;
end

%-----
function [TwMax,VwMax] = BoundaryCheck(A,BOD,C,d,gamma,N,n,Tsm,Vm)
% Evaluate V/Vm and T/Tm around the channel edge to see how close they to the boundary condition of zero

% Determine coordinates of channel edge to a 0.1 deg resolution
theta = (0:(1/N)/450:1/N)' ; % Angle/pi rad
r = ChannelEdge(BOD,d,gamma,N,theta);

% Calculate V/Vm and T/Tm around channel edge
j = 0:n;

```

```

S = r.^(j*N).*cospi(j*N.*theta); % NB Notice the use of 'cospi' - angle supplied as multiple of pi rad
Vw = (r.^2/4 - S*A)/Vm; % Actually, -V/Vw, but as only want maximum deviation from zero, this does not matter
Tw = (r.^4/64 - (r.^(j*N+2)).*cospi(j*N.*theta)./(4*j*N+4))*A - S*C)/Tsm;

% Maximum deviation of V/Vm and T/Tm on channel edge from zero
VwMax = max(abs(Vw));
TwMax = max(abs(Tw));

end

%-----
function r = ChannelEdge(BOD,d,gamma,N,theta)
% Returns r coordinate of edge of channel for required angular coordinate (theta)
% NB theta is angle as multiple of pi, i.e. angle in radians divided by pi, while BOD is in rad

IsStraight = theta*pi <= BOD; % True if boundary point lies on straight side, false if on round side
r = IsStraight.* d./(2*cospi(theta)) + ~IsStraight.* d/2.*((1-gamma)*sqrt(2).*cospi(1/N-theta) + sqrt(gamma.^2 - 2*(1-
gamma).^2.*sinpi(1/N-theta).^2)); % NB Notice the use of 'cospi' - angle supplied as multiple of pi rad

end

%-----
function [A,C] = EvalConstants(BOD,d,gamma,n,N,m)
% Evaluate constants in algebraic-trigonometric series from boundary conditions using the least squares method

% Validate input
if m<=(n+1), error('Number of boundary points (m) must be greater than number of terms in series (n+1).'); end

% Determine angles for boundary points, i.e. points where evaluate BC. BCs are located at equal angle interval
theta = (0:(1/N)/(m-1):1/N)'; % Angle/pi rad. Angles are evenly spaced

% Determine r for points on channel wall where evaluate boundary condition
r = ChannelEdge(BOD,d,gamma,N,theta);

% Set up matrices
D = r.^2/4; % Matrix containing r^2/4 terms
E = r.^4/64; % Matrix containing r^4/64 terms
j = 0:n;
S = r.^(j*N).*cospi(j*N.*theta); % Matrix containing terms for summation for each boundary point. NB1 Used for momentum and energy
balance. NB2 Notice the use of 'cospi' - angle supplied as multiple of pi rad
U = r.^(j*N+2).*cospi(j*N.*theta)./(4*j*N+4); % Matrix containing terms for summation for each boundary point for energy balance

% Solve for constants in matrices A and C
% A = (S'*S) \ S'*D; % This is the obvious implementation using left divide. However, it results in 'matrix is close to singular or badly
scaled' warnings with high values of n
% C = (S'*S) \ S'*(E - U*A);
A = pinv(S)*D; % Use of pinv avoids any warnings
C = pinv(S)*(E - U*A);
end

```

```

%-----
function [Ac,BOD,dH,p2_divAc] = GeoCalc(d,gamma)
% Geometric calculations for square channel with rounded corner

% Angle BOD: value of theta at which straight side stops and the round side starts
BOD = atan(1 - gamma); % Angle in radians

% Cross-sectional area of channel
Ac = d^2 *(1 - (1 - pi/4)*gamma^2);

% (Channel perimeter)^2/(Channel cross section)
p2_divAc = (4 - (4 - pi)*gamma)^2/(1 - (1 - pi/4)*gamma^2);

% Calculate hydraulic diameter
dH = d*(1 - (1 - pi/4)*gamma^2)/(1 - (1 - pi/4)*gamma);

end

%-----
function [alpha,alphaE,Tsm,Vm] = VelocityTempCalc(A,Ac,BOD,C,d,gamma,mAng,N,n)
% Calculate mean velocity (Vm), momentum flux correction factor (alpha), kinetic energy flux correction factor (alphaE), & normalised mixing cup
temperature (Tsm)

% Calculate mean velocity (Vm), alpha (<V^2>/<V>^2), alphaE (<V^3>/<V>^3) and Tsm (T*m in paper)
% Actually, Vm is not mean velocity, but (mean velocity)(viscosity)/(-dP/dz)
theta = (0:(1/N)/(mAng-1):1/N)'; % Angles used for numeric integration. Actually, angle/pi rad
r = ChannelEdge(BOD,d,gamma,N,theta); % r values for channel edge
j = 0:n;
aV = r.^4/16 - (r.^(j*N+2).*cospi(j*N.*theta)./(j*N+2))*A; % NB Notice the use of 'cospi' - angle supplied as multiple of pi
rad
aS = r.^6/96 - 1/2 *(r.^(j*N+4).*cospi(j*N.*theta)./(j*N+4))*A;
eS = r.^8/512 - 3/16*(r.^(j*N+6).*cospi(j*N.*theta)./(j*N+6))*A;
TS = r.^8/2048 - 1/64*((j*N+5).*r.^(j*N+6).*cospi(j*N.*theta)./((j*N+1).*(j*N+6)))*A - 1/4*(r.^(j*N+4).*cospi(j*N.*theta)./(j*N+4))*C;
for k = 0:n
    aS = aS + A(k+1)*cospi(k*N*theta) .* (r.^(j*N+2).*cospi(j*N.*theta)./((j+k)*N+2))*A;
    eS = eS + 3/4*A(k+1)*cospi(k*N*theta) .* (r.^(j*N+4).*cospi(j*N.*theta)./((j+k)*N+4))*A;
    TS = TS + (A(k+1)*r.^(j*N+4)./(4*(j*N+1).*(j+k)*N+4)) + C(k+1)*r.^(j*N+2)./(j+k)*N+2).*cospi(j*N.*theta).*cospi(k*N*theta))*A;
    for l = 0:n
        eS = eS - A(k+1)*A(l+1)*cospi(k*N*theta).*cospi(l*N*theta) .* (r.^(j*N+2).*cospi(j*N.*theta)./(j+k+1)*N+2))*A;
    end
end
Vm = trapz(theta,-aV)*pi *2*N/(Ac ); % trapz does trapezium rule integration. Actually, Vm is not mean velocity, but (mean
velocity)(viscosity)/(-dP/dz)
alpha = trapz(theta, aS)*pi *2*N/(Ac*Vm^2);
alphaE = trapz(theta,-eS)*pi *2*N/(Ac*Vm^3);
Tsm = trapz(theta,-TS)*pi *2*N/(Ac*Vm );
end

```

```

%-----
function [T,V,x,y] = VelocityTempField(A,BOD,C,d,gamma,mG,N,n,Tsm,Vm)
% Calculate velocity (V) and temperature (T) profiles over Cartesian grid

% Calculate V/Vm and T* as a function of position with Cartesian coordinates
[x,y] = meshgrid(linspace(0,d/2,mG)); % Create x-y grid for 1/4 of channel
r = sqrt(x.^2 + y.^2) ; % Convert to polar coordinates - r
theta = atan(y./x) ; % Convert to polar coordinates - angle in rad
theta(1,1) = 0 ; % Replace nan value - actual value not important as r=0 at this point
V = -r.^2/4 ; % When calculated, this is not actually velocity, but (velocity)(viscosity)/(-dP/dz)
T = r.^4/64 ;
for j = 0:n
    S = r.^(j*N).*cos(j*N.*theta);
    V = V + A(j+1).*S;
    T = T - A(j+1).*r.^(j*N+2).*cos(j*N.*theta)./(4*j*N+4) - C(j+1).*S;
end
V = V/Vm ; % Calculate: velocity/(mean velocity)
T = T/Tsm; % Non-dimensionalised temperature, i.e. (T - Tw)/(Tm - Tw)

% Set velocities and temperatures outside of channel to nan
IsRound = theta>BOD & theta<(pi/2-BOD); % True if angle corresponds to round/curved side
ii = IsRound & r>(d/2.*((1-gamma)*sqrt(2).*cos(pi/N-theta) + sqrt(gamma.^2 - 2*(1-gamma)^2.*sin(pi/N-theta).^2)));
V(ii) = nan; % If r larger than distance to channel edge at this angle, set to nan. NB Only need to do this for round sides; straight sides lie
on side of plot
T(ii) = nan;

end

%-----
function WriteConsts(gamma,X,Xstr)
% Writes out constants (i.e. elements of A or C) to Command Window

fprintf('g = %.3f\t%s =',gamma,Xstr);
fprintf('\t%f',X);
fprintf('\n');

end
%-----

```